(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2001－125829
（P2001－125829A）
(43)公開日　平成13年５月11日(2001.5.11)

(51)Int.Cl.'	識別記号	ＦＩ		テーマコード'（参考）
Ｇ０６Ｆ　12/08		Ｇ０６Ｆ　12/08	Ｄ	５Ｂ００５
			Ｓ	５Ｂ０１３
	３２０		３２０	５Ｂ０３３
3/06	３０２	3/06	３０２Ａ	５Ｂ０６５
9/32	３１０	9/32	３１０Ｊ	

審査請求　未請求　請求項の数20　ＯＬ　（全 16 頁）　最終頁に続く

(21)出願番号　特願平11－306605

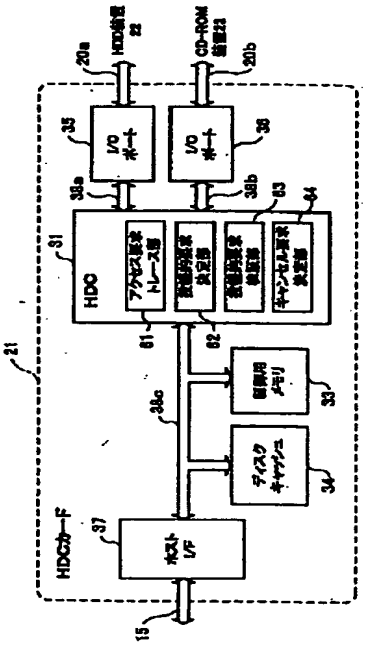
(22)出願日　平成11年10月28日(1999.10.28)

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国10504、ニューヨーク州　アーモンク　（番地なし）
(74)代理人　100104880
弁理士　古部　次郎　（外１名）

最終頁に続く

(54)【発明の名称】　コントローラ装置、ディスクコントローラ、補助記憶装置、コンピュータ装置、および補助記憶装置の制御方法

(57)【要約】
【課題】　アプリケーションからの真の要求に対して外部コントローラにより詳細な解析・予測を行い、ＨＤＤ等の補助記憶装置に対して先読み要求を発行する。
【解決手段】　データを記憶すると共にキャッシュメモリを有するＨＤＤ装置２２に接続され、このＨＤＤ装置２２を制御するＨＤＣカード２１であって、ホストにて実行されるアプリケーションプログラムからなされる真のアクセス要求を、アプリケーションプログラムから直接トレースするアクセス要求トレース部６１と、トレースされた真のアクセス要求に基づいて今後予想される投機的要求を決定する投機的要求決定部６２と、決定された投機的要求をＨＤＤ装置２２に対して発行するＨＤＣ３１とを備える。



【特許請求の範囲】

【請求項１】　データを記憶する補助記憶装置と当該補助記憶装置に対してアクセス要求を出すホスト装置との間に設けられ、当該補助記憶装置を制御するコントローラ装置であって、
前記ホスト装置から要求された過去のアクセス要求を記憶するアクセス要求記憶手段と、
前記アクセス要求記憶手段により記憶された過去のアクセス要求に基づいて、その後にアクセス要求されると予想されるデータの先読み要求を前記補助記憶装置に対して出力する先読み要求出力手段と、
前記先読み要求出力手段により出力した前記先読み要求の中から特定の先読み要求をキャンセルするためのキャンセル信号を前記補助記憶装置に対して出力するキャンセル信号出力手段と、を備えたことを特徴とするコントローラ装置。

【請求項２】　前記先読み要求出力手段により出力されるデータの先読み要求は、直ぐにその場で実行されるノン・キュー要求または前記補助記憶装置に一旦は待ち行列の形で保持されるタグ・キュー要求であることを特徴とする請求項１記載のコントローラ装置。

【請求項３】　前記キャンセル信号出力手段から出力されるキャンセル信号は、前記タグ・キュー要求に対してはキャンセルしたいタグ番号が指定されたキャンセル信号であることを特徴とする請求項２記載のコントローラ装置。

【請求項４】　前記キャンセル信号出力手段から出力されるキャンセル信号は、ＡＴＡ関連インターフェイスにおけるＮＯＰコマンドを拡張し、引数にキャンセルしたいタグ番号を指定したコマンドであることを特徴とする請求項３記載のコントローラ装置。

【請求項５】　実行中のコマンドに対して前記キャンセル信号出力手段から出力されるキャンセル信号は、ＡＴＡ関連インターフェイスにおけるデバイス・コントロール・レジスタの空きビットを用い、ソフトリセットを発行する形にて実行中のコマンドだけを中断させる信号であることを特徴とする請求項２記載のコントローラ装置。

【請求項６】　データを記憶すると共にキャッシュメモリを有するディスク状記憶装置に接続され、当該ディスク状記憶装置を制御するディスクコントローラであって、
ホストにて実行されるアプリケーションプログラムから前記ディスク状記憶装置に対してなされる真のアクセス要求を、当該アプリケーションプログラムから直接トレースするアクセス要求トレース部と、
前記アクセス要求トレース部によりトレースされた真のアクセス要求に基づいて今後予想される投機的要求を決定する投機的要求決定部と、
前記投機的要求決定部により決定された投機的要求を前記ディスク状記憶装置に対して発行するアクセス要求発行部と、を含むことを特徴とするディスクコントローラ。

【請求項７】　前記アクセス要求発行部から発行された投機的要求の中からキャンセルすべき特定の投機的要求を決定するキャンセル要求決定部と、
前記キャンセル要求決定部により決定された特定の投機的要求に対し、前記ディスク状記憶装置に対してキャンセル指示を発行するキャンセル指示発行部とを更に備えたことを特徴とする請求項６記載のディスクコントローラ。

【請求項８】　前記アクセス要求発行部から発行される投機的要求は、前記ディスク状記憶装置の媒体から直ぐにデータ読み出しを実行する要求であり、前記キャンセル指示発行部より発行されるキャンセル指示は、実行中の前記要求を中断させる指示であることを特徴とする請求項７記載のディスクコントローラ。

【請求項９】　前記アクセス要求発行部から発行される投機的要求は、タグ番号を指定して前記ディスク状記憶装置の内部に待ち行列として保持されるコマンドの要求であり、前記キャンセル指示発行部より発行されるキャンセル指示は、前記待ち行列中の要求の中から特定のタグ番号に該当する要求をキャンセルする指示であることを特徴とする請求項７記載のディスクコントローラ。

【請求項１０】　データを記憶する記憶媒体と、
予測される読み出し要求が実行され、前記記憶媒体からデータを読み出して一時的に蓄積するキャッシュメモリと、
タグ番号の情報を含む前記キャッシュメモリへの予測読み出し要求を外部コントローラ側から受信するインターフェイスと、
前記インターフェイスにより受信した予測読み出し要求に対し、前記タグ番号により当該要求が識別された待ち行列の状態にて、要求実行前の複数の要求を保持するコントローラとを備え、
前記インターフェイスは、要求の実行をキャンセルすべき特定タグ番号を指定したキャンセル信号を前記外部コントローラから受信し、
前記コントローラは、前記キャンセル信号を解析すると共に、指定された前記特定タグ番号に対応する要求を待ち行列から削除することを特徴とする補助記憶装置。

【請求項１１】　前記インターフェイスは、実行中の要求がキャンセルされた場合に実行中のデータのどこまでが有効データかを前記外部コントローラ側に対して送信することを特徴とする請求項１０記載の補助記憶装置。

【請求項１２】　アプリケーションプログラムを実行するホストと、
キャッシュメモリおよび内部コントローラを有すると共に、前記ホストからのアクセス要求に基づいてデータをリード・ライトする外部記憶装置と、

前記外部記憶装置を制御するコントローラとを備え、
前記コントローラは、前記ホストから出力された前記アクセス要求に基づいてその後にアクセス要求されると予想されるデータの先読み要求を前記外部記憶装置に対して出力すると共に、当該先読み要求をキャンセルするためのキャンセル信号を前記外部記憶装置に対して出力することを特徴とするコンピュータ装置。

【請求項１３】　前記先読み要求は前記外部記憶装置の記憶媒体から前記キャッシュメモリに対して直ぐにデータ読み出しを実行する要求であり、前記キャンセル信号は当該要求により実行中のコマンドを中断させる信号であることを特徴とする請求項１２記載のコンピュータ装置。

【請求項１４】　前記先読み要求は識別番号を付した状態で前記外部記憶装置の前記内部メモリによって管理されるコマンドの要求であり、前記キャンセル信号は当該内部メモリによって管理されるコマンドの中からキャンセルすべき特定の識別番号を指定する信号であることを特徴とする請求項１２記載のコンピュータ装置。

【請求項１５】　前記外部記憶装置は、前記コントローラから出力された前記キャンセル信号を解析して先読み要求をキャンセルすると共に、前記キャッシュメモリに書き込まれたデータのどこまでが有効データかを示す最終有効データの情報を前記コントローラに対して出力することを特徴とする請求項１２記載のコンピュータ装置。

【請求項１６】　アプリケーションからのコマンド発行を受け取るステップと、
前記アプリケーションから受けたコマンドの流れを解析するステップと、
解析された前記コマンドの流れに基づいて、先行して読み出すべきデータを指示する投機コマンドが必要か否かを判断するステップと、
投機コマンドが必要と判断された場合には、補助記憶装置に対して前記投機コマンドを発行するステップと、
それまでに発行した投機コマンドを検証するステップと、
検証結果によって不要な発行済み投機コマンドが存在すると判断される場合には、前記補助記憶装置に対してキャンセルコマンドを発行するステップと、を含むことを特徴とする補助記憶装置の制御方法。

【請求項１７】　前記投機コマンドは、前記補助記憶装置の媒体から直ぐにデータ読み出しを実行すべき要求を含むものであり、
前記キャンセルコマンドは、前記投機コマンドを実行中に当該投機コマンドを中断させることを特徴とする請求項１６記載の補助記憶装置の制御方法。

【請求項１８】　前記投機コマンドは、前記補助記憶装置の内部に待ち行列として保持される要求を含むものであり、
前記キャンセルコマンドは、前記投機コマンドにより保持された待ち行列中のコマンドの中から特定のコマンドを選定して当該待ち行列から削除することを特徴とする請求項１６記載の補助記憶装置の制御方法。

【請求項１９】　前記投機コマンドは、タグ番号を指定して保持される要求を含むものであり、
前記キャンセルコマンドは、キャンセルしたいタグ番号を指定することを特徴とする請求項１８記載の補助記憶装置の制御方法。

【請求項２０】　前記補助記憶装置からキャンセル処理終了後の最終有効データの情報を受信するステップと、を更に具備したことを特徴とする請求項１６記載の補助記憶装置の制御方法。

【発明の詳細な説明】

【０００１】
【発明の属する技術分野】本発明は、補助記憶装置を用いたコンピュータ装置または制御方法に関し、特にキャッシュメモリを用いてシステム全体のパフォーマンスの向上を図るための装置または方法に関する。

【０００２】
【従来の技術】近年のコンピュータシステムでは、ＣＰＵやシステム・メモリ、ＨＤＤ(ハードディスクドライブ：Hard Disk Drive)に代表される補助記憶装置(外部記憶装置)等により構成され、これらはデータ転送速度がそれぞれ異なるデバイスによって構成されている。これらのデバイスによって構成されるコンピュータシステムの性能を考えたときに、ＣＰＵやシステム・メモリのデータ転送速度スピードに比べて、ＨＤＤをはじめとする補助記憶装置のスピードがはるかに遅い。一般に、システム全体のデータ転送速度は、その速度の遅いデバイスによって支配されることから、これらの補助記憶装置の遅れがシステム全体におけるシステム性能のボトルネックとなっているのが現実である。

【０００３】これらのデバイス間の速度差を緩衝して処理速度を向上させるために、コンピュータシステムでは、一般に、キャッシュメモリが設けられている。このキャッシュメモリは、例えばＣＰＵと主記憶装置との間に置かれ、一度ＣＰＵが使用したデータや命令を、主記憶装置より高速なキャッシュメモリへ保存し、２回目からはこのキャッシュメモリから読み込み、主記憶装置へは直接入出力を行わないようにすることによって処理の高速化を図るものである。

【０００４】一方、補助記憶装置(外部記憶装置)であるＨＤＤでも磁気ディスク(メディア：媒体)の一部のデータを保持するキャッシュメモリが設けられ、予測される要求データを予めこのキャッシュメモリに入れておくことで、基本性能の向上と時間のかかる媒体へのアクセスを最小限に抑えるように構成されている。このように要求されているデータが予めキャッシュメモリに入っていれば、ＨＤＤのアクセス時間はシステム全体から見ても

性能のボトルネックになることはない。

【０００５】ここで、このような要求されるデータを予測する方法として、“Look Ahead”と呼ばれる先読みによるデータ読み出しのアルゴリズムが、最も一般的に採用されている。この“Look Ahead”は、要求されたデータの先(要求されたアドレスよりも大きいアドレス)のデータを予め読み出す技法、即ち、ホスト装置から要求される要求データの領域読み出しが終了した後に、その要求データに続く領域のデータも併せて読み出す技法である。この“Look Ahead”と呼ばれる先読みアルゴリズムは、それまでの要求パターンにかかわらず、常にその先のデータを読み込んでおくという、その処理に必要なオーバーヘッドを最小限に抑えた方法である。この方法を用いて真にパフォーマンスを向上させるためには、アプリケーションやＯＳからの要求パターンを細かく解析し、それ以降の要求を正確に予測することが理想的である。

【０００６】

【発明が解決しようとする課題】一方、ＨＤＤに代表される補助記憶装置では、どのようなコマンドが過去に発行されたかのトレース(足跡)を取り、次にどこに対してアクセス要求があるかを簡単に予測している。そして、内部にあるキャッシュメモリに予想されるデータを前もって入れておくことでパフォーマンスを上げるように構成されている。ところが、例えばＨＤＤに対して発行される要求は、アプリケーションやＯＳが真に要求したものではなく、実際には、ハードディスクコントローラ(ＨＤＣ)等がいくつかのアクセスをまとめたり、並べ直したりして、変更されたパターンにて投げられた要求であった。その結果、ＨＤＤで真の要求パターンを解析することができず、読み出しのキャッシュヒット率を上げることが困難であった。

【０００７】また、ＨＤＤ等の補助記憶装置における内部コントローラでは、本来の業務である、媒体へのリード・ライトアクセスを制御しながらホストからのコマンドをトレースする必要がある。その為に、アプリケーションやＯＳからの要求パターンを細かく解析し、分析することは、この内部コントローラによる負荷があまりにも大きすぎ、現実には、細かな解析や分析を行うことは困難であった。

【０００８】本発明は、以上のような技術的課題を解決するためになされたものであって、その目的とするところは、ＨＤＤ等の補助記憶装置とＨＤＣ等の外部コントローラとを協業させ、システム全体のパフォーマンスを向上させることにある。また他の目的は、アプリケーションやＯＳから発行された真の要求に基づいて外部コントローラにより詳細な解析・予測を行い、真の「予測」に基づく「投機的」な要求を前もって補助記憶装置に発行することにある。更に他の目的は、実行中または実行前の要求をキャンセルできる仕組みを持つことで、予測が外れたときのパフォーマンスの劣化を最小限に抑えることにある。

【０００９】

【課題を解決するための手段】かかる目的のもと、本発明のコントローラ装置は、データを記憶する補助記憶装置とこの補助記憶装置に対してアクセス要求を出すホスト装置との間に設けられ、この補助記憶装置を制御するコントローラ装置であって、ホスト装置から要求された過去のアクセス要求を記憶するアクセス要求記憶手段と、このアクセス要求記憶手段により記憶された過去のアクセス要求に基づいて、その後にアクセス要求されると予想されるデータの先読み要求を補助記憶装置に対して出力する先読み要求出力手段と、この先読み要求出力手段により出力した前記先読み要求の中から特定の先読み要求をキャンセルするためのキャンセル信号を補助記憶装置に対して出力するキャンセル信号出力手段と、を備えたことを特徴としている。

【００１０】このコントローラ装置の形態としては、ＰＣまたはホストとハードディスク装置等の補助記憶装置との間に、例えばハードディスクコントローラカード(ＨＤＣカード)を設ける形態が考えられる。また、ＰＣまたはホスト内部にその機能を設け、実質的にホスト装置と補助記憶装置との間にコントローラ装置が設けられる形態も含まれるものである。補助記憶装置の内部に設けられた内部コントローラとは区別されるものであれば、その形態が問われるものではない。また、この先読み要求出力手段により出力されるデータの先読み要求は、直ぐにその場で実行されるノン・キュー要求または補助記憶装置に一旦は待ち行列の形で保持されるタグ・キュー要求であることを特徴とすれば、予測されるコマンドに応じて最適な要求を補助記憶装置に送信することができる点で好ましい。特に、このキャンセル信号出力手段から出力されるキャンセル信号は、タグ・キュー要求に対してはキャンセルしたいタグ番号が指定されたキャンセル信号であることを特徴とすれば、実行中の任意のコマンドをキャンセルすることが可能となり、システム全体のパフォーマンスの向上が図れる点で優れている。

【００１１】また、このキャンセル信号出力手段から出力されるキャンセル信号は、ＡＴＡ関連インターフェイスにおけるＮＯＰコマンドを拡張し、引数にキャンセルしたいタグ番号を指定したコマンドであることを特徴とすることもできる。更に、実行中のコマンドに対してキャンセル信号出力手段から出力されるキャンセル信号は、ＡＴＡ関連インターフェイスにおけるデバイス・コントロール・レジスタの空きビットを用い、ソフトリセットを発行する形にて実行中のコマンドだけを中断させる信号であることを特徴とすることもできる。このＡＴＡ(AT Attachment)関連インターフェイスとしては、関連するＡＴＡＰＩ(ＡＴＡ Packet Interface)などの拡

張プロトコル等を含むものである。

【００１２】また、本発明は、データを記憶すると共にキャッシュメモリを有するディスク状記憶装置に接続され、このディスク状記憶装置を制御するディスクコントローラであって、ホストにて実行されるアプリケーションプログラムからディスク状記憶装置に対してなされる真のアクセス要求を、アプリケーションプログラムから直接トレースするアクセス要求トレース部と、このアクセス要求トレース部によりトレースされた真のアクセス要求に基づいて今後予想される投機的な要求を決定する投機的要求決定部と、この投機的要求決定部により決定された投機的要求をディスク状記憶装置に対して発行するアクセス要求発行部と、を含むことを特徴としている。

【００１３】この「真のアクセス要求」は、アクセス要求がまとめられたり、並べ直されたりしていないアプリケーションプログラムからのそのままの要求であれば、予測の正確性を各段に上昇させることができる点で好ましい。また、アクセス要求トレース部は、アプリケーションプログラムからなされた複数の要求をその順序情報と共に記憶し、アクセス要求の足跡を把握できる形態にあることが好ましい。尚、このディスクコントローラの形態としては、ディスク状記憶装置の内部に設けられる内部コントローラと区別されるものであれば、その適用形態が問われるものではない。この内部コントローラと区別されることで、ディスク状記憶装置の内部コントローラで行われていた業務の一部を移管し、更に優れたアクセス要求パターンの解析を行うことが可能となる点で優れている。

【００１４】更に、アクセス要求発行部から発行された投機的要求の中からキャンセルすべき特定の投機的要求を決定するキャンセル要求決定部と、このキャンセル要求決定部により決定された特定の投機的要求に対し、ディスク状記憶装置に対してキャンセル指示を発行するキャンセル指示発行部とを更に備えたことを特徴とすれば、実行中または実行前の要求を必要に応じてキャンセルすることが可能となり、予測が外れたときのパフォーマンスの劣化を最小限に留めることができる点で優れている。

【００１５】このアクセス要求発行部から発行される投機的要求は、ディスク状記憶装置の媒体から直ぐにデータ読み出しを実行する要求であり、このキャンセル指示発行部より発行されるキャンセル指示は、実行中の要求を中断させる指示であることを特徴とすれば、実行中の要求であっても速やかにキャンセルすることでパフォーマンスの向上を図ることが可能である。また、このアクセス要求発行部から発行される投機的要求は、タグ番号を指定してディスク状記憶装置の内部に待ち行列として保持されるコマンドの要求であり、キャンセル指示発行部より発行されるキャンセル指示は、待ち行列中の要求の中から特定のタグ番号に該当する要求をキャンセルする指示であることを特徴とすれば、複数の要求を「待ち行列」に入れることができると共に、予測が外れたときでも、その「待ち行列」から外すことで実行前の要求に対する不要なアクセスを防止することができる点で好ましい。

【００１６】また、本発明の補助記憶装置は、データを記憶する記憶媒体と、予測される読み出し要求が実行され、この記憶媒体からデータを読み出して一時的に蓄積するキャッシュメモリと、タグ番号の情報を含むこのキャッシュメモリへの予測読み出し要求を外部コントローラ側から受信するインターフェイスと、このインターフェイスにより受信した予測読み出し要求に対し、タグ番号によりこの要求が識別された待ち行列の状態にて、要求実行前の複数の要求を保持するコントローラとを備え、このインターフェイスは、要求の実行をキャンセルすべき特定タグ番号を指定したキャンセル信号を外部コントローラから受信し、コントローラは、キャンセル信号を解析すると共に、指定された特定タグ番号に対応する要求を待ち行列から削除することを特徴としている。

【００１７】ここで、複数の要求を「待ち行列」に入れることで、コントローラは、補助記憶装置内部で実行順序の効率等を考慮した上で順番を変えて要求を実行することが可能となる。更に、実行前の要求を簡易にキャンセルすることができ、外部コントローラによる予測が外れた場合であってもパフォーマンスの劣化を最小限に抑えることが可能となる。また、このインターフェイスは、実行中の要求がキャンセルされた場合に実行中のデータのどこまでが有効データかを外部コントローラ側に対して送信することを特徴とすれば、外部コントローラ側にて、次に送信すべき予測読み出し要求を正確に決定できる点で優れている。

【００１８】また、本発明が適用されたコンピュータ装置は、アプリケーションプログラムを実行するホストと、キャッシュメモリおよび内部コントローラを有すると共に、ホストからのアクセス要求に基づいてデータをリード・ライトする外部記憶装置と、この外部記憶装置を制御するコントローラとを備え、このコントローラは、ホストから出力されたアクセス要求に基づいてその後にアクセス要求されると予想されるデータの先読み要求を外部記憶装置に対して出力すると共に、この先読み要求をキャンセルするためのキャンセル信号を外部記憶装置に対して出力することを特徴としている。

【００１９】また、この先読み要求は外部記憶装置の記憶媒体からキャッシュメモリに対して直ぐにデータ読み出しを実行する要求であり、キャンセル信号はこの要求により実行中のコマンドを中断させる信号であることを特徴とすることができる。更に、この先読み要求は識別番号を付した状態で外部記憶装置の内部メモリによって管理されるコマンドの要求であり、キャンセル信号はこ

の内部メモリによって管理されるコマンドの中からキャンセルすべき特定の識別番号を指定する信号であることを特徴とすることができる。また更に、この外部記憶装置は、コントローラから出力されたキャンセル信号を解析して先読み要求をキャンセルすると共に、キャッシュメモリに書き込まれたデータのどこまでが有効データかを示す最終有効データの情報をこのコントローラに対して出力することを特徴とすれば、キャンセルした要求の一部のデータをコントローラが有効に利用できる点で好ましい。

【００２０】また、この発明に係る補助記憶装置の制御方法は、アプリケーションからのコマンド発行を受け取るステップと、このアプリケーションから受けたコマンドの流れを解析するステップと、解析されたこのコマンドの流れに基づいて、先行して読み出すべきデータを指示する投機コマンドが必要か否かを判断するステップと、この投機コマンドが必要と判断された場合には、補助記憶装置に対して投機コマンドを発行するステップと、それまでに発行した投機コマンドを検証するステップと、検証結果によって不要な発行済み投機コマンドが存在すると判断される場合には、補助記憶装置に対してキャンセルコマンドを発行するステップと、を含むことを特徴としている。

【００２１】この投機コマンドは、補助記憶装置の媒体から直ぐにデータ読み出しを実行すべき要求を含むものであり、このキャンセルコマンドは、投機コマンドを実行中にこの投機コマンドを中断させることを特徴とすれば、現在実行中の要求に対しても適切にキャンセルすることが可能となる点で好ましい。また、この投機コマンドは、補助記憶装置の内部に待ち行列として保持される要求を含むものであり、このキャンセルコマンドは、投機コマンドにより保持された待ち行列中のコマンドの中から特定のコマンドを選定して待ち行列から削除することを特徴とすることができる。更に、この投機コマンドは、タグ番号を指定して保持される要求を含むものであり、このキャンセルコマンドは、キャンセルしたいタグ番号を指定することを特徴とすれば、「待ち行列」として補助記憶装置内に保持されるタグ・キュー(Tagged Queue)要求に対し、タグ番号の指定による簡単な指示により投機コマンドのキャンセルをすることが可能となる。また、補助記憶装置からキャンセル処理終了後の最終有効データの情報を受信するステップとを更に具備したことを特徴とすれば、例えば既にアクセスしたデータを有効に活用でき、システム全体のパフォーマンスの向上を図ることができる点で優れている。

【００２２】

【発明の実施の形態】実施の形態１

図１は、本実施の形態１におけるコンピュータ装置(コンピュータシステム)の概略構成を示す図である。同図に示すように、このコンピュータ装置は、ホスト(ＨＯＳＴ)側１０として、ＣＰＵ１２、第１ブリッジ回路１３およびメインメモリ１４を備えている。このＣＰＵ１２は、外部バス１１ａを介して第１ブリッジ回路１３に接続されている。また、第１ブリッジ回路１３は、外部バス１１ｂを介してメインメモリ１４が接続されている。また第１ブリッジ回路１３には、周辺装置を拡張する際に用いられる拡張バス１５が接続されている。この第１ブリッジ回路１３には、例えば、図示しないＣＰＵインターフェイス、メモリコントローラ、クロック発生器、ＰＣＩバス(Peripheral Component Interconnect bus)インターフェイス等の機能が搭載されている。アプリケーションプログラムは、メインメモリ１４に格納されたプログラムに基づいてＣＰＵ１２により実行される。また、メインメモリ１４には、デバイスドライバのプログラムが格納されており、このデバイスドライバはアプリケーションプログラムとは独立して、後述するＨＤＤ装置２２への読み書きの指示を行っている。

【００２３】拡張バス１５には、ＨＤＣ(Hard Disk Controller)カード２１が接続されている。また、ＨＤＣカード２１には、バス２０ａを介して補助記憶装置(外部記憶装置)としてのＨＤＤ(Hard Disk Drive)装置２２が接続されており、更に、バス２０ｂを介してＣＤ-ＲＯＭ装置２３が接続されている。このＨＤＣカード２１は、ＨＤＤ装置２２やＣＤ-ＲＯＭ装置２３に対して外部コントロール装置として機能しており、特に、ＨＤＤ装置２２の制御機能の拡張や、他のＨＤＤ装置の増設時に用いられる。また、ＨＤＤ装置２２には、ディスクキャッシュ５４が設けられており、予測されるデータを予めキャッシュしておき、時間のかかる媒体へのアクセスを最小限に抑えるように構成されている。更に、本実施の形態では、バス２０ａはＡＴＡ(AT Attachment)バスにより構成されており、バス２０ｂは例えばＡＴＡ/ＡＴＡＰＩ(ATA Packet Interface)バスにより構成されている。

【００２４】図２は、ＨＤＤ装置２２の概略構成を示す図である。同図に示されるように、ＨＤＤ装置２２は、駆動機構として、記憶媒体としてデータを記憶する磁気ディスク４１、磁気ディスク４１を回転駆動するスピンドルモータ４２を備えている。磁気ヘッド４３は、磁気ディスク４１へのデータの記録・再生(リード・ライト)を行う。ヘッドアーム４４は、その先端に磁気ヘッド４３を備え、磁気ディスク４１の記録面の上空を移動している。また、アクチュエータ４５は、ヘッドアーム４４を保持すると共にヘッドアーム４４を回転駆動させている。これらによって、磁気ヘッド４３は、磁気ディスク４１の略半径方向を移動し、磁気ディスク４１の記録面における任意の位置にてアクセスできるように構成されている。

【００２５】磁気ディスク４１、スピンドルモータ４２、磁気ヘッド４３およびアクチュエータ４５から構成

される駆動機構は、制御回路５０によって制御されている。この制御回路５０は、ＨＤＣ(Hard Disk Controller)５１、制御用メモリ５３、ディスクキャッシュ５４およびホストＩ/Ｆ５５を備えており、これらはバス５６を介して相互に接続されている。

【００２６】ＨＤＣ５１は、ＨＤＤ装置２２の内部コントローラとして、制御用メモリ５３に記憶された制御プログラムと制御データに従ってＨＤＤ装置２２の全体を制御している。このＨＤＣ５１は、サーボ制御やデータのリード・ライト時の誤り制御の為の演算処理を実行している。これによって、スピンドルモータ４２やアクチュエータ４５を駆動し、磁気ヘッド４３の記録用ヘッドおよび再生用ヘッドを用いてリード・ライトを実行している。また、ＨＤＣ５１は、記憶媒体としての磁気ディスク４１に記録されるデータの一部をディスクキャッシュ５４に保持し、また、磁気ディスク４１に格納されているデータの一部を先読みしてディスクキャッシュ５４に保持するための制御を行っている。

【００２７】制御用メモリ５３には、ＨＤＣ５１により実行される制御プログラムおよびこの制御プログラムにより使用される制御データが格納されている。ディスクキャッシュ５４は、磁気ディスク４１に記録される書き込みデータを一時的に記憶すると共に、磁気ディスク４１から読み出された読み出しデータを一時的に記憶するためのキャッシュメモリとしての機能を有する。このディスクキャッシュ５４は、例えばＤＲＡＭから構成されており、数Ｍ～数十ＭＢの記憶容量を備えている。また、ホストＩ/Ｆ５５は、ＨＤＣカード２１との間でデータやコマンドを送受するインターフェイス回路である。

【００２８】このホストＩ/Ｆ５５は、外部コントローラであるＨＤＣカード２１から先読みデータに関する情報として、今後読み出しが予測される複数の種類からなる要求を受信する。これらの要求は、直ぐにその場で実行するための要求であるノン・キュー(Non Queue)要求と、直ぐには実行されないタグを付したタグ・キュー(Tagged Queue)要求とに大別される。ノン・キュー要求を受け取ると、ＨＤＣ５１は、直ぐに磁気ディスク４１からの読み出しを開始し、読み出したデータをディスクキャッシュ５４に格納する。一方、タグ・キュー要求を受け取ると、ＨＤＣ５１は、複数の要求を一旦、「待ち行列」という形で保持する。この「待ち行列」は、例えばレーテンシィが少なくなるような順番にて内部的に要求を実行できるように順番が決定される。保持された複数の要求は、このように実行順序の効率等が考慮され、順番を変えた上で読み出しが実行される。更に、ホストＩ/Ｆ５５は、ＨＤＣカード２１からキャンセル要求を受信する。また、作成された実行中のコマンドをキャンセルしたときにＨＤＣ５１により有効データが作成されるが、この有効データは、ホストＩ/Ｆ５５を介してＨＤＣカード２１に発行される。

【００２９】図３は、ＨＤＣカード２１の概略構成を示す図である。ＨＤＣカード２１は、ＨＤＣ３１、制御用メモリ３３、ディスクキャッシュ３４、Ｉ/Ｏポート３５、３６およびホストＩ/Ｆ３７を備えており、ＨＤＤ装置２２の外部コントローラとして機能している。このＨＤＣ３１、制御用メモリ３３、ディスクキャッシュ３４、およびホストＩ/Ｆ３７は、バス３８ｃを介して相互に接続されている。Ｉ/Ｏポート３５はバス３８ａを介してＨＤＣ３１に接続されており、また、Ｉ/Ｏポート３６はバス３８ｂを介してＨＤＣ３１に接続されている。このＨＤＣ３１は、制御用メモリ３３に記憶されている制御プログラムおよび制御データに基づいてＨＤＣカード２１の全体を制御している。また、ＨＤＣ３１は、ＨＤＤ装置２２に格納されるデータの一部をディスクキャッシュ３４に保持するための制御を実施している。

【００３０】制御用メモリ３３には、ＨＤＣ３１により実行される制御プログラムと、この制御プログラムに使用される制御データが格納されている。ディスクキャッシュ３４は、ＨＤＤ装置２２に記憶される書き込みデータの一時記憶と、ＨＤＤ装置２２から読み出されたデータの一時記憶を行うキャッシュメモリとしての機能を備えている。このディスクキャッシュ３４は、図２に示すＨＤＤ装置２２に設けられたディスクキャッシュ５４に対して上位のキャッシュメモリに相当し、例えばＤＲＡＭにより構成されて数Ｍ～数十ＭＢの記憶容量を有している。更に、Ｉ/Ｏポート３５には、バス２０ａを介して補助記憶装置としてのＨＤＤ装置２２が接続され、Ｉ/Ｏポート３６には、バス２０ｂを介してＣＤ-ＲＯＭ装置２３が接続されている。また、ホストＩ/Ｆ３７は、拡張バス１５と接続されており、ホスト側と接続されてデータを送受するインターフェース回路である。

【００３１】ＨＤＣカード２１は、コンピュータ装置(コンピュータシステム)の起動時にＨＤＤ装置２２との間で設定情報を送受し、送受された設定情報に基づいてＨＤＣカード２１に初期設定を行う。また、本実施の形態では、従来、ＨＤＤ装置２２単体で行っていたホストからの要求データの予測を、ＨＤＣカード２１により実行するようにして、予測の精度を上げ、更に要求パターンを細かく解析して詳細な予測を可能とするように構成されている。そのために、ＨＯＳＴ側１０のアプリケーションから発せられるＨＤＤ装置２２に対する実行要求を、まとめや並び替え等が行われていない、そのままの形、即ち、「真のアクセス要求」を、ホストＩ/Ｆ３７から受信するように構成されている。更に、Ｉ/Ｏポート３５からは、ＨＤＤ装置２２に対して真の「予測」に基づく投機的な要求が発行されると共に、後述するキャンセル信号が発せられるように構成されている。

【００３２】ここで、ＨＤＣ３１の機能をブロック的に

表現する。図３に示すように、ＨＤＣ３１は、アクセス要求トレース部６１、投機的要求決定部６２、投機的要求検証部６３、およびキャンセル要求決定部６４とを備えている。アクセス要求トレース部６１は、アプリケーションから発行されたコマンド(アクセス要求)を受けて、その要求を記憶し、更に、そのコマンドの足跡、流れを解析してトレースしている。投機的要求決定部６２は、アクセス要求トレース部６１にてトレースされた要求を解析し、次に要求が来るであろうコマンドを能動的に決定している。投機的要求検証部６３は、既にＨＤＤ装置２２に対して発行された投機的(speculation)要求(投機コマンド)を蓄積して解析、検証している。また、キャンセル要求決定部６４は、投機的要求検証部６３による検証結果を受けて、既に発行され且つキャンセルすべき投機的要求を決定するように構成されている。この決定を受けてキャンセルコマンドが発行される。

【００３３】次に、本実施の形態における投機コマンドおよびキャンセルコマンド発行の流れについて、図１～図４を用いて説明する。図４は、アプリケーションプログラム、ＨＤＣカード２１およびＨＤＤ装置２２の３者でのやりとりを、ＨＤＣカード２１の動きから説明するものである。まず、ＨＤＣカード２１のＨＤＣ３１は、ＨＯＳＴ側１０のアプリケーションからのコマンドの発行を待つ(ステップ１０１)。アプリケーションからのコマンドの発行を検出すると(ステップ１０２)、それまでのアプリケーションからのコマンドの流れを解析する(ステップ１０３)。即ち、ＨＤＣ３１のアクセス要求トレース部６１は、複数受けた真のアクセス要求のコマンドを常時モニターして予測を立てておく。尚、ステップ１０２にて，アプリケーションからのコマンドの発行がなされない場合には、ステップ１０１のアプリケーションからのコマンドの発行を待つ。ステップ１０３の後、投機的要求決定部６２では、投機コマンドが必要かどうかを判断する(ステップ１０４)。この予測としては、例えば、幾つかのアクセス要求パターンを格納しておき、トレースした真のアクセス要求がこれらの格納パターンと一致するか否かを判断する方法がある。本実施の形態では、予測手法については従来の技術を採用すれば足り、例えば、“Look Ahead”と呼ばれる先読みによるデータ読み出しのアルゴリズムを用いた予測を用いることもできる。このような予測により投機コマンドの発行が必要になると、ＨＤＣ３１は、Ｉ/Ｏポート３５を経てバス２０ａを介し、ＨＤＤ装置２２に対して投機コマンドを発行する(ステップ１０５)。尚、ステップ１０４にて投機コマンドが必要でないと判断された場合には、そのまま、後述するステップ１０６に移る。

【００３４】ＨＤＤ装置２２に対して投機コマンドを発行したＨＤＣ３１は、投機的要求検証部６３にて、それまでに発行した投機コマンドの検証を行う(ステップ１０６)。キャンセル要求決定部６４は、この検証によって不要な発行済み投機コマンドが存在するか否かを判断する(ステップ１０７)。不要な発行済み投機コマンドが存在しない場合には、ステップ１０１であるアプリケーションからのコマンドの発行待ちに戻る。不要な発行済み投機コマンドが存在する場合には、Ｉ/Ｏポート３５を経てバス２０ａを介し、ＨＤＤ装置２２に対してキャンセルコマンドを発行する(ステップ１０８)。その後、ステップ１０１であるアプリケーションからのコマンドの発行待ちに戻り、今まで説明した、投機コマンドおよびキャンセルコマンド発行の流れを繰り返すように構成されている。

【００３５】ここで、投機コマンドとしては、前述のようにノン・キュー要求とタグ・キュー要求が発行されるが、ノン・キュー要求の投機コマンドについては、ＨＤＤ装置２２にて直ぐに読み出しが実行される。そのために、バス２０ａが使用された状態でキャンセルコマンドを発行する必要がある。本実施の形態では、これら実行中のキャンセル要求に対しては、所謂ソフト・リセットと同様な機能を用いてＨＤＤ装置２２に対してキャンセルをかけるように構成されている。また、一方で、実行中の要求がキャンセルされた場合に、実行中のデータのどこまでが有効データかを把握することができれば、キャンセルした要求の一部のデータを有効利用できる点で好ましい。かかる点を考慮して、本実施の形態では、ＨＤＤ装置２２によるキャンセル処理後、空きレジスタに最終有効データのＬＢＡ(LogicalBlock Address)を、ＨＤＣカード２１に対して返すように構成し、ＨＤＣ３１にてどこまでが有効データかを認識できる方法を確立している。

【００３６】また、タグ・キュー要求が発行された後、ＨＤＤ装置２２により実際に実行された要求は、ＨＤＣ３１にて常に認識された状態にある。投機的要求検証部６３では、これらの実行済みの要求を踏まえて投機コマンドを検証している。実行前のタグ・キュー要求をキャンセルする場合には、タグ番号を指定してキャンセルすることで、ＨＤＤ装置２２で「待ち行列」に入っている複数の要求から、特定のタグ・キュー要求をキャンセルすることが可能となる。また、タグ・キュー要求であっても現在実行中の要求に関しては、ノン・キュー要求におけるキャンセルと同様な方法にてキャンセルをかけることが可能である。この際には、どこまでが有効データかの情報が、前述と同様に、ＨＤＤ装置２２からＨＤＣカード２１に対して発行される。

【００３７】次に、本実施の形態にて用いるコマンドについて、具体例を用いて詳述する。図５(ａ)、(ｂ)は、本実施の形態で用いるＡＴＡインターフェイス規格におけるリード･コマンドの一例を示している。図５(ａ)はノン・キュー(Non Queue)要求、図５(ｂ)はタグ・キュー(Tagged Queue)要求に使用するリードＤＭＡ(Direct Memory Access)コマンドである。コマンドブロックに

は、フィーチャ、セクタ･カウント、セクタ･ナンバー、シリンダ・ロー、シリンダ・ハイ、デバイス/ヘッドおよびコマンドの各レジスタがあり、これらは全て８ビット幅である。ＡＴＡの全ての動作は、コマンドレジスタにコマンドコードを書き込むことによって実行される。本実施の形態では、ＬＢＡ(Logical Block Address)方式を採用することから、デバイス/ヘッド・レジスタの上位４ビットのうち、上から２ビット目を“1”(High)としてＬＢＡ方式を選択している。そのために、セクタ･カウント、シリンダ・ローおよびシリンダ・ハイの各レジスタによる3バイトと、デバイス/ヘッド・レジスタの下位４ビットによってＬＢＡを指定する。

【００３８】図6(ａ)、(ｂ)、(ｃ)は、本実施の形態で用いるＡＴＡインターフェイス規格におけるレジスタおよび規格を拡張したコマンドを示している。本実施の形態では、これらのレジスタおよびコマンドを用いて、既に発行した投機コマンドをキャンセルするように構成している。ここで、図6(ａ)は８ビットのデバイス・コントロール・レジスタであり、下位３ビット目の“ｂ２”にＳＲＳＴ＝１を書き込むと、全てのデバイスがリセットされる。本実施の形態では、所定の空きビットを用いて、実行中のコマンドだけをキャンセルするように構成している。また図6(ｂ)は本実施の形態のために改良されたＮＯＰコマンドである。従来のＮＯＰコマンドでは、全てのキュー・コマンドをキャンセルすることは可能であったが、特定コマンドだけをキャンセルさせる指示を発行することはできなかった。そこで、本実施の形態では、セクタ・カウント・レジスタの上位５ビットである、ビット７からビット３を用いて、キューからキャンセルしたいタグ番号を指定できるように構成し、不要なアクセス要求だけをＨＤＤ装置２２にキャンセルさせることを可能とした。また、図6(ｃ)は、図6(ｂ)に示すＮＯＰコマンドのフィーチャ(Features)・レジスタの内容を示したものである。コード“００ｈ”では、全てのコマンドをキャンセルするアクションを示している。また、コード“０２ｈ”では、指定したタグをキャンセルするアクションを示している。更に、他のコードでは、何もしないアクションを示している。

【００３９】次に、本実施の形態を用いたアクセスの一例について、図７～図９を用いて説明する。ここで、図７はアクセスの一例における概要を示す説明図である。また、図８は、この図７のアクセス例でノン・キュー要求のケースにおけるコマンドの一例を説明するための図である。また、図９は、図７のアクセス例でタグ・キュー要求のケースにおけるコマンドの一例を説明するための図である。

【００４０】まず、図７を用いてアクセス例を説明する。図７では、最初に、■アプリケーションから[ＬＢＡ：０から１０ブロック]の読み出し要求、次に、■アプリケーションから[ＬＢＡ：１００から１０ブロック]の要求、次に、■アプリケーションから[ＬＢＡ：１０から１０ブロック]の要求があった場合を示している。図３に示すＨＤＣ３１のアクセス要求トレース部６１が、この■■■の要求をトレースし、これらの要求に基づいてＨＤＣ３１の投機的要求決定部６２が、次に[ＬＢＡ：１１０から]の要求が来ることを予測した。この予測に基づいて、■投機コマンド[ＬＢＡ：１１０から５０ブロック]を決定した状態を示している。また、予測に反して■アプリケーションから[ＬＢＡ：２０から１０ブロック]の読み出し要求があった場合を示している。

【００４１】この図７におけるアクセス例で、ＨＤＣカード２１からＨＤＤ装置２２に発行されるノン・キュー要求のコマンド例を、図８を用いて説明する。図8(ａ)は、図７の、■アプリケーションから[ＬＢＡ：０から１０ブロック]の要求があった場合、ＨＤＤ装置２２に発行されるＡＴＡコマンド(図5(ａ))である。ＬＢＡレジスタのビットは、全て“0”であり、セクタ・カウント・レジスタは、ブロック数“１０”を示している。また、コマンド・レジスタは、“Ｃ８ｈ”で、「リトライ有り」を示している。図8(ｂ)では、■の要求、即ち、アプリケーションから[ＬＢＡ：１００から１０ブロック]の要求があり、ＬＢＡ“１００”、セクタ・カウント“１０”がＨＤＤ装置２２に発行される。図8(ｃ)では、■の要求、即ち、アプリケーションから[ＬＢＡ：１０から１０ブロック]の要求があり、ＬＢＡ“１０”、セクタ・カウント“１０”がＨＤＤ装置２２に発行される。

【００４２】図8(ｄ)は、アプリケーションからの■■■の要求に基づき、次に[ＬＢＡ：１１０から]の要求が来ることを予測して、[ＬＢＡ：１１０から５０ブロック]の要求を■の投機コマンドとしてＨＤＤ装置２２に発行した場合を示している。ＬＢＡ“１１０”、セクタ・カウント“５０”がＨＤＤ装置２２に発行される。これらのコマンド例から明らかなように、ＨＤＤ装置２２では、真のアクセス要求に基づくコマンドか、投機コマンドか、の区別は行われない。図8(ｅ)は、予測に反してアプリケーションから[ＬＢＡ：２０から１０ブロック]の要求があったので、■の要求をキャンセルするコマンドをＨＤＤ装置２２に発行している。ここでは、図6(ａ)に示したデバイス・コントロール・レジスタの例えば下位４ビット目である(bit３)を１にしている。このように本実施の形態では、デバイス・コントロール・レジスタの空きビットを使用し、処理としてはコマンドの実行中にソフト・リセットを発行するイメージで、実行中のコマンドを中断させている。これにより、図３に示すバス２０ａが使用されている状態でも、ＨＤＤ装置２２に対し、実行中のコマンドを中断させることが可能となる。

【００４３】この実行中のコマンドが中断された場合

に、本実施の形態では、ＨＤＤ装置２２からＨＤＣカード２１に対して、実行されたデータのどこまでが有効データかを知らせるように構成している。より具体的には、例えば、図５(ａ)に示したノン・キュー要求に使用するリード・コマンドと同様なインターフェイスを用い、そのレジスタに最終有効データのＬＢＡを挿入して、ＨＤＣカード２１に対して通知すれば良い。このように構成することで、キャンセルした要求の一部のデータをＨＤＣカード２１が有効に利用することができ、システム全体のパフォーマンスを向上させることが可能となる。尚、インターフェイスは任意に選択することが可能であり、任意の空きレジスタに最終有効データのＬＢＡを付するような方法にて、有効データのレポートを行うことが可能である。その後、図８(ｆ)に示すように、上記のアプリケーションの要求■[ＬＢＡ：２０から１０ブロック]が、ＨＤＤ装置２２に対して発行される。

【００４４】次に、図７におけるアクセス例で、ＨＤＣカード２１からＨＤＤ装置２２に発行されるタグ・キュー要求のコマンド例を、図９を用いて説明する。図９(ａ)は、図７の、■アプリケーションから[ＬＢＡ：０から１０ブロック]の要求があった場合、ＨＤＤ装置２２に発行されるＡＴＡコマンド(図５(ｂ))である。フィーチャ・レジスタでは、セクタ・カウント数として、ブロック数“１０”を示している。また、セクタ・カウント・レジスタの上位５ビット(ビット３～ビット７)を用いてタグ番号を示しており、ビット３を１とした“０８ｈ”によってタグ番号１を示している。ＬＢＡレジスタのビットは、全て“０”である。また、コマンド・レジスタは、“Ｃ７ｈ”で、このコマンドがリードＤＭＡ・キュー・コマンドであることを示している。図９(ｂ)では、■の要求、即ち、アプリケーションから[ＬＢＡ：１００から１０ブロック]の要求があり、フィーチャ“１０”、タグ番号を示すセクタ・カウントがビット４を１とした“１０ｈ”によってタグ番号２を示し、ＬＢＡ“１００”がＨＤＤ装置２２に発行される。図９(ｃ)では、■の要求、即ち、アプリケーションから[ＬＢＡ：１０から１０ブロック]の要求があり、フィーチャ“１０”、セクタ・カウント“１８ｈ”によってタグ番号３、ＬＢＡ“１０”がＨＤＤ装置２２に発行される。

【００４５】図９(ｄ)は、アプリケーションからの■■■の要求に基づき、次に[ＬＢＡ：１１０から]の要求が来ることを予測して、[ＬＢＡ：１１０から５０ブロック]の要求を■の投機コマンドとしてＨＤＤ装置２２に発行した場合を示している。フィーチャ“５０”、セクタ・カウント“２０ｈ”によってタグ番号４、ＬＢＡ“１１０”が発行される。このコマンドを受けたＨＤＤ装置２２は、真のアプリケーションからの要求か、ＨＤＣカード２１によって生成された投機コマンドなのか、を区別することがなく、同様な要求として「待ち行列」に保持している。

【００４６】図９(ｅ)は、予測に反してアプリケーションから[ＬＢＡ：２０から１０ブロック]の要求があったので、■の要求をキャンセルするコマンドをＨＤＤ装置２２に発行している。ここでは、図６(ｂ)に示した拡張ＮＯＰコマンドを用いて指示している。具体的には、フィーチャ・レジスタを、図６(ｃ)に示す“０２ｈ”とし、このコマンドが、キューを選択してキャンセルする要求であることを、特記事項として示している。また、キャンセルすべきキューの指定として、セクタ・カウント・レジスタを用いて“２０ｈ”のタグ番号４を示している。また、コマンド・レジスタは、このコマンドがＮＯＰコマンドであることを示す“００ｈ”が送信される。その後、図９(ｆ)に示すように、上記のアプリケーションの要求■[ＬＢＡ：２０から１０ブロック]を、セクタ・カウント“２８ｈ”でタグ番号５を示し、ＨＤＤ装置２２に対して発行される。

【００４７】尚、タグ・キュー要求を行ったキューをキャンセルする方法としては、コマンド実行中であれば、その実行中のコマンドを中断させるように構成することもできる。この場合には、図８(ｅ)にて説明したノン・キュー要求のキャンセルと同様な方法で、ソフトリセットを発行するイメージにて実行中のコマンドを中断させることが可能である。一方、実行中のコマンドが中断された場合には、前述と同様に、最終有効データのＬＢＡをＨＤＣカード２１にレポートすれば、キャンセルまでの実行データを有効に利用できる点で優れている。このときのレポートの仕方は、前述の方法と同様な方法を採用すれば良い。

【００４８】このように、本実施の形態によれば、ＨＤＤ装置２２が今まで行っていた処理の一部をＨＤＣカード２１が分業し、ＨＤＣカード２１によって、アプリケーションからの真の要求をもとに、高度な、且つ詳細な予測を立てることが可能となる。特に、他のＨＤＤ装置を増設して予測の正確性を増す場合に、拡張ボード等で構成されるＨＤＣカード２１にこれらの予測とコマンド発行機能を設けることで、機能拡張の際に最小限の変更にて、高度なコントロールが可能となる。

【００４９】実施の形態２

実施の形態１では、例えば拡張ボードに代表されるＨＤＣカード２１を用いてＨＤＤ装置２２へのコマンド発行を制御したが、実施の形態２では、ＨＯＳＴ側にこのＨＤＣカード２１の機能を備え、例えばマザーボードに設けられたメインメモリにＨＤＤ装置２２へのコマンド発行制御機能を備えたものである。尚、実施の形態１と同様の機能については、同様の符号を用い、ここでは、その詳細な説明を省略する。

【００５０】図１０は、実施の形態２におけるコンピュータ装置(コンピュータシステム)の概略構成を示す図である。同図に示すように、このコンピュータ装置では、ＨＯＳＴ側７０における拡張バス１５に第２ブリッジ回

路７１が接続されている。この第２ブリッジ回路７１には、バス７５を介して図示しない周辺回路が接続可能である。また、この第２ブリッジ回路７１には、バス７４ａを介してＨＤＤ装置２２が接続され、バス７４ｂを介してＣＤ-ＲＯＭ装置２３が接続されている。バス７５は、例えばＩＳＡ(Industry Standard Architecture)バスから構成されている。また、本実施の形態では、バス７４ａはＡＴＡバスにより構成されており、バス７４ｂは例えばＡＴＡ/ＡＴＡＰＩバスにより構成されている。第２ブリッジ回路７１は、拡張バス１５と規格が異なる拡張バスに周辺装置を接続するために設けられている。第２ブリッジ回路７１は、ＰＣＩバスインターフェイス、ＩＳＡバスインターフェイス、システムＩ/Ｏコントローラ、ＤＭＡコントローラ等の機能を有している。

【００５１】メインメモリ８０は、ＣＰＵ７９の命令に従って、ＨＤＤ装置２２との間でデータの授受を行う。このメインメモリ８０には、プログラムであるデバイスドライバ８１が格納されており、デバイスドライバ８１は、アプリケーションプログラムとは独立して、ＨＤＤ装置２２への読み書きの指示を行っている。このデバイスドライバ８１には、その機能として、アプリケーションからのＨＤＤ装置２２に対するアクセス要求を解析してトレースするアクセス要求トレース部８２を有する。また、アクセス要求トレース部８２にてトレースされた要求を解析し、次に要求が来るであろうコマンドを能動的に決定する投機的要求決定部８３を備える。また、投機的要求決定部８３による決定に基づいて、投機的要求(投機コマンド)をＨＤＤ装置２２に発行した後に、投機的要求を蓄積して解析する投機的要求検証部８４を有している。更に、この投機的要求検証部８４による検証結果に基づいて、既に発行され且つキャンセルすべき投機的要求を決定するキャンセル要求決定部８５を備えている。

【００５２】これらの機能を有するデバイスドライバ８１は、実施の形態１にて説明したＨＤＣカード２１のＨＤＣ３１と同様に、ＨＤＤ装置２２に対して投機コマンドとキャンセルコマンドを発行する。即ち、デバイスドライバ８１から発せられたこれらのコマンドは、第１ブリッジ回路１３、第２ブリッジ回路７１を経て、バス７４ａを介してＨＤＤ装置２２に対して発行される。ＨＤＤ装置２２では、実施の形態１と同様に、これらのコマンドを解析する。タグ・キュー要求に対しては、「待ち行列」としてその内部にアクセス要求を保持し、キャンセル要求に基づいて指示されたタグのキューをキャンセルする。また、実行中のコマンドに対してキャンセルコマンドが発行された場合には、直ぐに実行を中断すると共に、デバイスドライバ８１に対して実行中のどこまでが有効データかを示すＬＢＡを返すように構成されている。このように、実施の形態２によれば、ＨＤＣカード２１等の拡張ボードを用いることなく、ホスト(ＨＯＳＴ)側にてＨＤＤ装置２２による制御を実行することができる。これにより、予め、ディスク装置(ＨＤＤ装置２２等)の拡張性を考慮して設計することが可能であれば、拡張ボードを別個、設ける態様に比べて、コストを低く抑えることが可能である。

【００５３】以上説明したように、本実施の形態１および２によれば、アプリケーションからの真の要求を解析してアクセス要求を解析することができるので、予測の正確性を増した「真の予測」を行うことができる。また、ＨＤＤ装置２２で行っていた処理の一部を分業する形で予測することから、ＨＤＤ装置２２の内部コントローラに大きな負荷をかけることなく、詳細な、且つ大掛かりな予測であっても実行することが可能となり、ＣＰＵにおけるパワーの有効利用と、予測の適合性を大きく高めることが可能となる。更に、本実施の形態によれば、ノン・キュー要求、タグ・キュー要求に分けてＨＤＤ装置２２に対してコマンドを発行することができると共に、これらのアクセス要求に対してキャンセルする仕組みを持つことが可能となる。

【００５４】

【発明の効果】以上説明したように、本発明によれば、アプリケーションからの真の要求に対して外部コントローラにより詳細な解析・予測を行い、ＨＤＤ等の補助記憶装置に対して先読み要求を発行することで、予測の正確性を増すことができる。また、実行中または実行前の要求を、外部コントローラからキャンセルできる仕組みを持つことで、予測が外れたときのパフォーマンスの劣化を最小限に抑え、システム全体のパフォーマンスを向上させることができる。

【図面の簡単な説明】

【図１】　本実施の形態１におけるコンピュータ装置(コンピュータシステム)の概略構成を示す図である。

【図２】　ＨＤＤ装置２２の概略構成を示す図である。

【図３】　ＨＤＣカード２１の概略構成を示す図である。

【図４】　本実施の形態における投機コマンドおよびキャンセルコマンド発行の流れを説明するためのフローチャートである。

【図５】　(ａ)、(ｂ)は、本実施の形態で用いるＡＴＡインターフェイス規格におけるリード･コマンドの一例を示す図である。

【図６】　(ａ)、(ｂ)、(ｃ)は、本実施の形態で用いるＡＴＡインターフェイス規格におけるレジスタおよび規格を拡張したコマンドの一例を示す図である。

【図７】　アクセスの一例における概要を示す説明図である。

【図８】　ノン・キュー要求におけるコマンドの一例を説明するための図である。

【図９】　タグ・キュー要求におけるコマンドの一例を

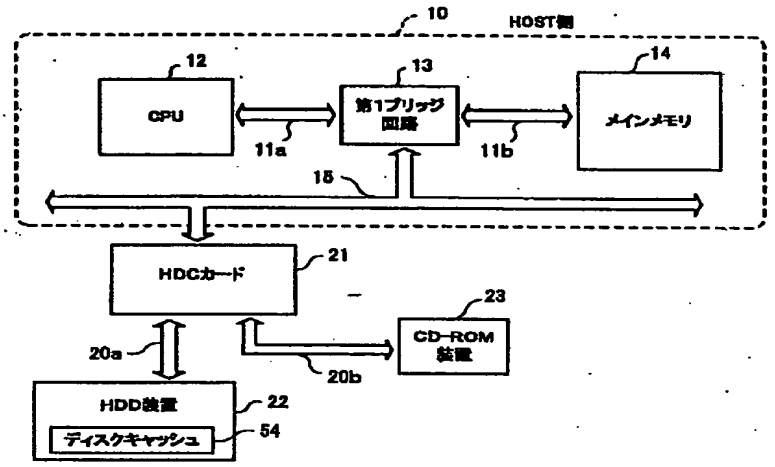
説明するための図である。

【図１０】　実施の形態２におけるコンピュータ装置(コンピュータシステム)の概略構成を示す図である。

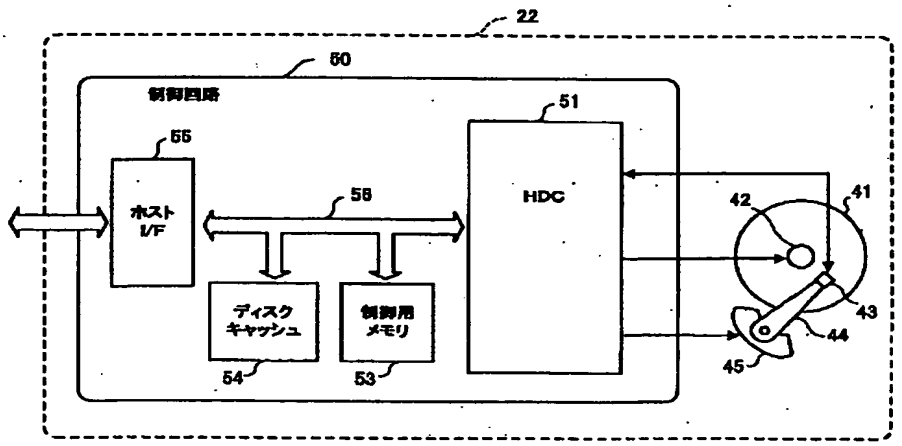
【符号の説明】

１０…ホスト(ＨＯＳＴ)側、１１ａ,１１ｂ…外部バス、１２…ＣＰＵ、１３…第１ブリッジ回路、１４…メインメモリ、１５…拡張バス、２０ａ,２０ｂ…バス、２１…ＨＤＣカード、２２…ＨＤＤ装置、３１…ＨＤＣ、３３…制御用メモリ、３４…ディスクキャッシュ、３５,３６…Ｉ/Ｏポート、３７…ホストＩ/Ｆ、３８ａ,３８ｂ,３８ｃ…バス、５０…制御回路、５３…制御用メモリ、５４…ディスクキャッシュ、５５…ホストＩ/Ｆ、６１…アクセス要求トレース部、６２…投機的要求決定部、６３…投機的要求検証部、６４…キャンセル要求決定部、７０…ＨＯＳＴ側、７１…第２ブリッジ回路、７４ａ,７４ｂ…バス、７９…ＣＰＵ、８０…メインメモリ、８１…デバイスドライバ、８２…アクセス要求トレース部、８３…投機的要求決定部、８４…投機的要求検証部、８５…キャンセル要求決定部

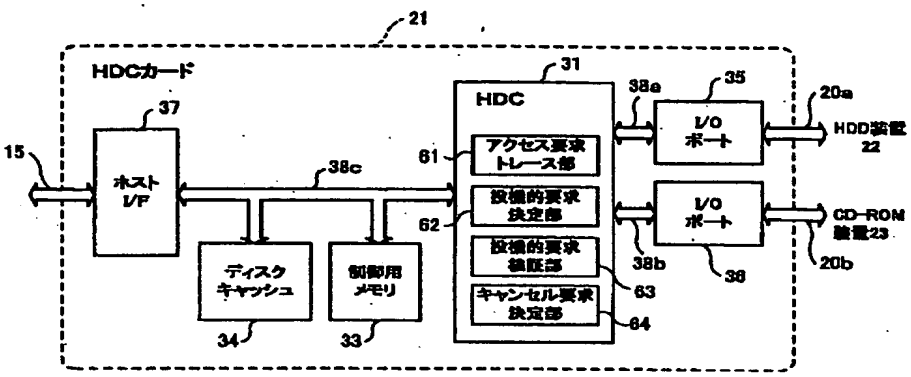
【図１】



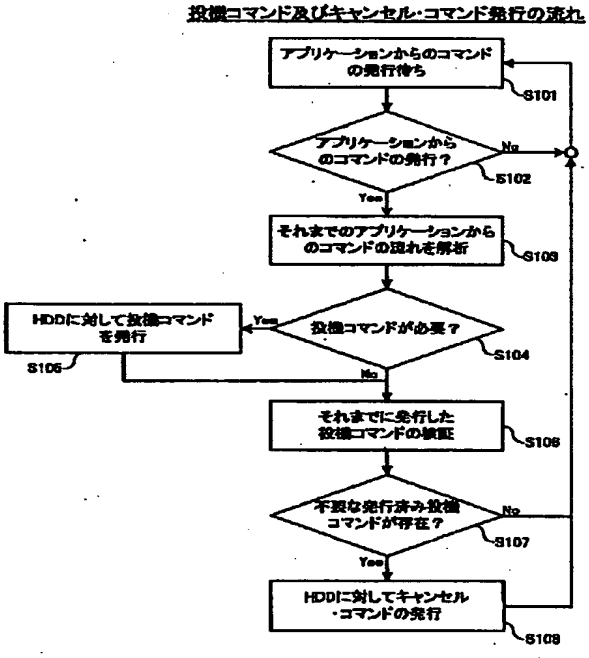
【図２】



【図３】



【図４】



【図５】

Non Queue要求に使用するRead command

Register	7	6	5	4	3	2	1	0
Features	na							
Sector Count	Sector count							
Sector Number	Sector number or LBA							
Cylinder Low	Cylinder low or LBA							
Cylinder High	Cylinder high or LBA							
Device/Head	obs	LBA	obs	DEV	Head number or LBA			
Command	C8h or C9h							

(a)

Tagged Queue要求に使用するRead command

Register	7	6	5	4	3	2	1	0
Features	Sector count							
Sector Count	Tag					na	na	na
Sector Number	Sector number or LBA							
Cylinder Low	Cylinder low or LBA							
Cylinder High	Cylinder high or LBA							
Device/Head	obs	LBA	obs	DEV	Head number or LBA			
Command	C7h							

(b)

【図６】

デバイス・コントロール・レジスタ

b7	b6	b5	b4	b3	b2	b1	b0
予約	予約	予約	予約	予約	SRST	nIEN	0

(a)

NOP command (00h)

Register	7	6	5	4	3	2	1	0
Features	Subcommand code							
Sector Count	Tag					na	na	na
Sector Number	Na							
Cylinder Low	Na							
Cylinder High	Na							
Device/Head	obs	na	obs	DEV	na	na	na	na
Command	00h							

(b)

Featuresレジスタの内容

Code	Description	Action
00h	NOP	全てのキューされているコマンドをキャンセルする。
01h	NOP Auto POLL	何もしない。
02h	NOP Selection POLL	指定したタグをキャンセルする。
03h-FFh	Reserved	何もしない。

(c)

【図９】

Tagged Queue要求のケース

Register	Contents
Features	10
Sector Count	08h(Tag:1)
LBA	0
Command	C7h

(a)

Register	Contents
Features	10
Sector Count	10h(Tag:2)
LBA	100
Command	C7h

(b)

Register	Contents
Features	10
Sector Count	18h(Tag:3)
LBA	10
Command	C7h

(c)

Register	Contents
Features	50
Sector Count	20h(Tag:4)
LBA	110
Command	C7h

(d)

Register	Contents
Feature	02h
Sector Count	20h(Tag:4)
Command	00h

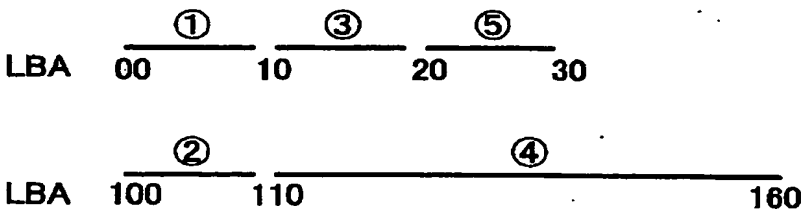
(e)

Register	Contents
Features	10
Sector Count	28h(Tag:5)
LBA	20
Command	C7h

(f)

【図７】

アクセスの概要



【図８】

Non Queue要求のケース

Register	Contents
LBA	0
Sector Count	10
Command	C8h

(a)

Register	Contents
LBA	100
Sector Count	10
Command	C8h

(b)

Register	Contents
LBA	10
Sector Count	10
Command	C8h

(c)

Register	Contents
LBA	110
Sector Count	50
Command	C8h

(d)

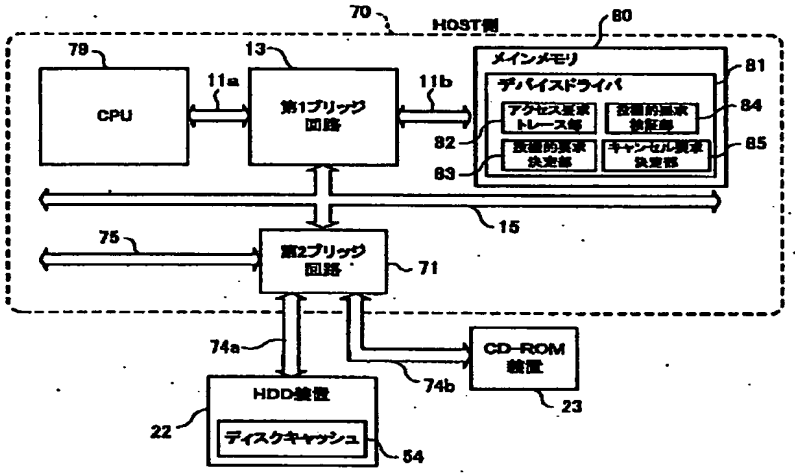
Register	Contents
Device Control	bit3 : 1

(e)

Register	Contents
LBA	20
Sector Count	10
Command	C8h

(f)

【図１０】



フロントページの続き

(51)Int.Cl.'		識別記号	FI		テーマコード'(参考)
G06F	9/34	350	G06F	9/34	350A
	9/38	310		9/38	310A

(71)出願人　599152728
プロミス・テクノロジー・インク
アメリカ合衆国　95112　カリフォルニア
サン・ホセ　コール・サークル　1460

(72)発明者　金丸　淳
神奈川県藤沢市桐原町１番地　日本アイ・ビー・エム株式会社　藤沢事業所内

(72)発明者　浅野　秀夫
神奈川県藤沢市桐原町１番地　日本アイ・ビー・エム株式会社　藤沢事業所内
(72)発明者　木橋　昭
神奈川県藤沢市桐原町１番地　日本アイ・ビー・エム株式会社　藤沢事業所内
(72)発明者　櫛田　弘一
神奈川県藤沢市桐原町１番地　日本アイ・ビー・エム株式会社　藤沢事業所内
(72)発明者　齋藤　高裕
神奈川県藤沢市桐原町１番地　日本アイ・ビー・エム株式会社　藤沢事業所内
(72)発明者　チチェン　ウー
アメリカ合衆国　94024　カリフォルニア　ロス・アルトス　ラーミー・プレイス　1150
(72)発明者　ケルビン　カオ
アメリカ合衆国　95135　カリフォルニア　サン・ホセ　ルーブル・アベニュー　4088

Ｆターム(参考)　5B005 JJ13 KK11 LL11 MM11 NN12 NN22 NN31
5B013 AA01
5B033 AA04 AA13 DB06
5B065 BA01 CH05

JAPAN PATENT OFFICE

PATENT LAID-OPEN OFFICIAL GAZETTE

Laid-Open No.
2001-125829

Laid-Open
H.13(2001) May 11

Application No.: H11-306605

Filed: H.11 (1999) Oct. 28

Applicant: 390009531
INTERNATIONAL BUSINESS MASCHINES CORPORATION
Armonk, New York 10504
United States

Attorney, Agent: Jiro Furube and another

(To be continued to the last page)

1. TITLE OF THE INVENTION

Controller Unit, Disk Controller, Auxiliary Storage Unit, Computer Unit, and a Control Method for Auxiliary Storage Unit

[Summary]

[Problem to be solved]

In response to a true request from an application, an external controller performs a detailed analysis and prediction, and issues read-ahead requests to auxiliary storage units, such as HDDs.

[Solution]

An HDC card 21 that is connected to an HDD unit 22 storing data and having cache memory and controls the HDD unit 22, the HDC card comprising: an access request tracing unit 61 that traces true access requests made by applications programs that are executed on a host through direct tracing from the applications programs, a speculative request determination unit 62 that determines speculative requests expected in the future based upon true access requests that are traced, and an HDC 31 that issues a speculative request that has been determined to an HDD unit 22.

2. WHAT IS CLAIMED IS:

[Claim 1]

A controller unit that is installed between an auxiliary storage unit storing data and a host unit that issues access requests to said auxiliary storage unit and that controls said auxiliary storage unit, the controller unit comprising: an access request storage means that

stores past access requests issued by the aforementioned host unit; a read-ahead request output means that outputs data read-ahead requests for the data that is expected to be subject to a subsequent access request based on the past access requests that are stored in the aforementioned access request storage means; and a cancellation signal output means that issues a cancellation signal to the aforementioned auxiliary storage unit that cancels a specific read-ahead request from among the read-ahead requests that were output by the aforementioned read-ahead request output means.

[Claim 2]

The controller unit of Claim 1, wherein the data read-ahead request issued by the aforementioned read-ahead request output means is either a non-queue request, which is executed immediately on the spot, or a tagged-queue request, which is temporarily held in the form of a queue.

[Claim 3]

The controller unit of Claim 2, wherein the cancellation signal that is output by the aforementioned cancellation signal output means is a cancellation signal in which a tag number of the aforementioned tagged-queue request that is to be cancelled is specified.

[Claim 4]

The controller unit of Claim 3, wherein the cancellation signal that is output by the aforementioned cancellation signal output means is a command which is an extended NOP command for the ATA-related interface and in which the tag number to be cancelled is specified as an argument.

[Claim 5]

The controller unit of Claim 2, wherein the cancellation signal that is output from the aforementioned cancellation signal output means for a command being

executed is a signal that only cancels the command being executed, in the form of issuing a soft-reset by using empty bits in the device control register in the ATA-related interface.

[Claim 6]

A disk controller connected to a disk-type storage device that stores data and has cache memory, and controlling said disk-type storage device, the disk controller comprising: an access request tracing unit that directly traces true access requests that are made by an applications program executed on a host to the aforementioned disk-type storage device from said applications program; a speculative request determination unit that determines speculative requests that are anticipated in the future, based upon the true access request traced by the access request tracing unit; and an access request issuing unit that issues the speculative request determined by the aforementioned speculative request determination unit to the aforementioned disk-type storage device.

[Claim 7]

The disk controller of Claim 6 further comprising: a cancellation request determination unit that determines a specific speculative request to be cancelled from among the speculative requests issued by the aforementioned access request issuing unit; and a cancellation instruction issuing unit that issues cancellation instructions to the aforementioned disk-type storage device.

[Claim 8]

The disk controller of Claim 7, wherein the speculative requests issued by the aforementioned access request issuing unit are requests for the immediate reading of data from the medium of the aforementioned disk-type storage device; and wherein the cancellation

instructions issued by the aforementioned cancellation instruction issuing unit are instructions for the cancellation of the aforementioned request being executed.

[Claim 9]

The disk controller of Claim 7, wherein the speculative request issued by the aforementioned access request issuing unit is a request, specifying a tag number, for a command that is held as a queue in the aforementioned disk-type storage device; and wherein the cancellation instruction issued by the aforementioned cancellation instruction issuing unit is an instruction that cancels the request matching a specific tag number among the requests stored in the aforementioned queue.

[Claim 10]

An auxiliary storage unit comprising: a storage medium in which data is stored; cache memory wherein anticipated read requests are executed and data is read from the aforementioned storage medium and temporarily accumulated therein; and an interface that receives anticipated read requests, including tag number information, to the aforementioned cache memory from an external controller; and a controller that holds a plurality of execution-pending requests in a queuing state in which the requests are identified by the aforementioned tag number, in response to the anticipated read requests received from the aforementioned interface; wherein the aforementioned interface receives from the aforementioned external controller a cancellation signal specifying a tag number indicating the specific request whose execution is to be canceled, and wherein the aforementioned controller parses the aforementioned cancellation signal, and deletes the request associated with the specified, specific tag number from the queue.

[Claim 11]

The auxiliary storage unit of Claim 10, wherein the aforementioned interface, if the request being executed is canceled, transmits to the aforementioned external controller information on what portion of the data being executed is valid.

[Claim 12]

A computer unit comprising: a host that executes applications programs, cache memory, and an internal controller, as well as external storage units that read and write data based on access requests from the aforementioned host and a controller that controls the aforementioned external storage units; wherein the aforementioned controller outputs to the aforementioned external storage unit a read-ahead request on the data for which a subsequent access request is anticipated to be made based on the aforementioned access request issued by the aforementioned host, and outputs to the aforementioned external storage unit a cancellation signal that cancels said read-ahead request.

[Claim 13]

The computer unit of Claim 12, wherein the aforementioned read-ahead request is a request for immediate execution of data read from the storage medium of the aforementioned external storage unit into the aforementioned cache memory; and wherein the aforementioned cancellation signal is a signal for the cancellation of the command being executed by virtue of said request.

[Claim 14]

The computer unit of Claim 12, wherein the aforementioned read-ahead request is a request for a command that is managed by the aforementioned internal memory under a condition in which an identification number is assigned to the command; and wherein the aforementioned cancellation signal is a signal that specifies the

specific identification number to be cancelled from among the commands that are managed by said internal memory.

[Claim 15]

The computer unit of Claim 12, wherein the aforementioned external storage unit analyzes the aforementioned cancellation signal output from the aforementioned controller, and wherein the external storage unit outputs to the aforementioned controller information on final effective data indicating what portion of the data written to the aforementioned cache memory is valid.

[Claim 16]

An auxiliary storage unit control method comprising: a step of receiving commands issued by an application; a step of analyzing the flow of commands received from the aforementioned application; a step of determining, based on the aforementioned flow of commands that has been analyzed, whether a speculative command that indicates the data to be read ahead is needed; a step of issuing the aforementioned speculative command to the auxiliary storage unit if it is determined that a speculative command is needed; a step of validating the speculative commands that have been issued; and a step of issuing a cancellation command to the aforementioned auxiliary storage unit if it is determined that, as a result of the validation, superfluously issued speculative commands are pending.

[Claim 17]

The auxiliary storage unit control method of Claim 16, wherein the aforementioned speculative command includes a request for the immediate execution of the reading of data from the medium of the aforementioned auxiliary storage unit, and wherein the aforementioned cancellation command cancels said speculative command when the aforementioned speculative command is being executed.

[Claim 18]

The auxiliary storage unit control method of Claim 16, wherein the aforementioned speculative command includes requests that are internally held as a queue in the aforementioned auxiliary storage unit, and wherein the aforementioned cancellation command selects a specific command from the commands held in the queue by the aforementioned speculative command and deletes it from said queue.

[Claim 19]

The auxiliary storage unit control method of Claim 18, wherein the aforementioned speculative command includes requests that are held through the specification of a tag number, and wherein the aforementioned cancellation command specifies a tag number to be canceled.

[Claim 20]

The auxiliary storage unit control method of Claim 16 further comprising: a step of receiving information on the last valid data from the aforementioned auxiliary storage unit upon completion of cancellation processing.

---

3. DETAILED DESCRIPTION OF THE INVENTION

[0001]

[Scope of Utilization in Industry]

This invention is directed to a computer unit or a control method using an auxiliary storage unit; in particular, it relates to a unit or a method designed to improve overall system performance through the use of cache memory.

[0002]
[Prior Art]

Modern computer systems are comprised of the CPU, the system memory, and auxiliary storage units (external storage units) represented by hard disk drives (HDDs). These units are comprised of devices with different data transfer rates. If we focus on a computer system comprised of these devices, HDDs and other auxiliary storage units have data transfer rates substantially slower than those of the CPU or the system memory. Given the fact that, as a general rule, the overall data transfer rate of a system is dominated by the slowest device in the system, the fact of the matter is that the slowness of these auxiliary storage units imposes a bottleneck on overall system performance.

[0003]

For the improvement of processing speed through the alleviation of speed differences among these devices, cache memory is generally provided in the computer systems. Cache memory, for example, is placed between the CPU and the main memory unit so that any data or instructions once used by the CPU are saved in the cache memory, which is faster than the main memory unit. In this manner, the processing speed is enhanced by reading data and instructions from the cache memory when they are accessed for the second time and more, so that no direct input/output operations are performed on the main memory unit.

[0004]

On the other hand, in the HDD, which is an auxiliary storage unit (external storage unit), cache memory is provided for holding part of the data on the magnetic disk, which improves the basic performance and minimizes time-consuming access to the media by storing data for which

access requests are anticipated in the cache memory. In this manner, placing requested data in the cache memory in advance prevents HDD access time from becoming a bottleneck on overall system performance.

[0005]

The most commonly adopted method for the prediction of data requested in this manner is the read-ahead data read algorithm called "Look Ahead", which is a technique of reading in advance data that is ahead (with an address greater than the requested data) of requested data. In other words, it is a technique of reading data lying in an area succeeding the requested data upon completion of the reading of the requested data area that is requested by a host device. The "Look Ahead" read-ahead algorithm is designed to minimize the overhead required for the processing by constantly reading data lying ahead, irrespective of previous patterns of requests. To truly improve performance by using this method, it would be ideal to analyze patterns of requests from the operating system and applications in detail to accurately predict any subsequent requests.

[0006]

[Problems to Be Solved by this Invention]

On the other hand, auxiliary storage units, such as HDDs, traces the pattern of what commands have been issued in the past, and in a simple manner predicts the target area of the next access request, and they are configured to improve performance by placing the expected data in the internal cache memory in advance. However, requests that are issued to an HDD are not genuine requests issued by applications and the operating system; instead, they are requests thrown in a modified pattern created by hard disk controllers (HDCs) through the grouping and sorting of several access requests. As a result, the HDD is unable to

analyze true request patterns, which makes the task of improving the read cache hit rate difficult.

[0007]

In addition, internal controllers in auxiliary storage units, such as HDDs, are required to trace commands from the host while controlling read/write access with respect to the media. Therefore, finely parsing and analyzing the patterns of requests from applications and the operating system imposes too great a burden on the internal controller, with the result that it has been difficult for it to perform parsing and analyses in detail.

[0008]

Having been developed to solve the technical issues described above, the objective of the present invention is to permit the coordination between auxiliary storage units, such as HDDs, and external controllers, such as HDCs, so as to achieve an improvement in overall system performance. Another objective of the present invention is to provide detailed parsing and analyses by an external controller, based on true requests issued by applications and the operating system so that "speculative" requests based on a true "prediction" can be issued to the auxiliary storage unit in advance. Yet another objective of the present invention is to provide a mechanism whereby requests being executed or requests pending execution can be canceled so as to minimize performance degradation in the event of a failed prediction.

[0009]

[Means of Solving the Problems]

Given these objectives, the controller unit of the present invention is installed between an auxiliary storage unit that stores data and a host unit that issues access requests to said auxiliary storage unit and controls the auxiliary storage unit; the controller unit

comprising an access request storage means that stores past access requests issued by the aforementioned host unit; aread-ahead request output means that outputs data read-ahead requests on the data that is expected to be subject to a subsequent access request based on the past access requests that are stored in the aforementioned access request storage means; and a cancellation signal output means that issues a cancellation signal to the aforementioned auxiliary storage unit that cancels a specific read-ahead request from among the read-ahead requests that were output by the aforementioned read-ahead request output means.

[0010]

A conceivable form of the controller unit is a hard disk controller card (HDC card), for example, installed between either a PC or a host and an auxiliary storage unit, such as a hard disk unit. Such a configuration also includes the form in which the required controller function is provided internal to the PC or the host so that in substance a controller unit is provided between the host unit and the auxiliary storage unit. The controller unit may be in any form as long as it is distinguishable from the internal controller that is provided internal to the auxiliary storage unit. In addition, the data read-ahead request issued by the read-ahead request output means should preferably be characterized by either a non-queue request, which is executed immediately on the spot, or a tagged-queue request, which is temporarily held in the form of a queue, so that the optimum request can be transmitted to the auxiliary storage unit according to the predicted command. In particular, the cancellation signal that is output by the cancellation signal output means should be characterized as a cancellation signal in which a tag number for canceling the aforementioned tagged-queue request is specified. Such a signal would be capable of

canceling any command being executed and offers the advantage of improving overall system performance.

[0011]

Also, the cancellation signal that is output by the cancellation signal output means can be characterized as a command that is an extended NOP command for the ATA-related interface and in which the tag number to be canceled is specified as an argument. Further, the cancellation signal that is output from the cancellation signal output means for the command being executed is a signal that only cancels the command being executed, in the form of issuing a soft-reset by using empty bits in the device control register in the ATA-related interface. Such an ATA (AT Attachment)-related interface includes extended protocols, such as related ATAPI (ATA Packet Interface).

[0012]

Further, the present invention is characterized as a disk controller connected to a disk-type storage device that stores data and has cache memory, and controlling said disk-type storage device, the disk controller comprising: an access request tracing unit that directly traces true access requests that are made by an applications program executed on a host to the aforementioned disk-type storage device from said applications program; a speculative request determination unit that determines speculative requests that are anticipated in the future, based upon the true access requests traced by the access request tracing unit; and an access request issuing unit that issues the speculative request determined by the aforementioned speculative request determination unit to the disk-type storage device.

[0013]

From the standpoint of further permitting a significant enhancement in the accuracy of prediction, it

would be desirable that the "true access request" be a request "as is" from the applications program, without the access request being grouped or re-sorted. In addition, it would be desirable that the access request tracing unit be in a form wherein it stores a plurality of access requests made by the applications program together with sequence information thereon, so that the access requests can be tracked. In terms of the form of the disk controller, the form of its application is immaterial provided it is distinguishable from an internal controller that is installed internal to the disk-type storage device. By being distinct from the internal controller, a part of the operations performed by the internal controller in the disk-type storage device can be delegated to the disk controller so that even further superior access request pattern analyses can be performed, which would be an advantage.

[0014]

The disk controller further comprises a cancellation request determination unit that determines a specific speculative request to be cancelled from among the speculative requests issued by the access request issuing unit; and a cancellation instruction issuing unit that issues cancellation instructions to the disk-type storage device. In this manner, the disk controller can cancel requests being executed or requests for which execution is pending, so that any performance degradation in the event of an off-target prediction can be minimized, which would be an advantage.

[0015]

The speculative requests issued by the access request issuing unit can be characterized as being requests for the immediate reading of data from the medium of the disk-type storage device; and the cancellation instructions issued by the cancellation instruction issuing unit can be instructions for the cancellation of the request being

executed. In this manner, requests, even when they are being executed, can be canceled rapidly, thereby improving performance. Further, the speculative request issued by the access request issuing unit is a request, specifying a tag number, for a command that is held as a queue in the disk-type storage device; and the cancellation instruction issued by the cancellation instruction issuing unit is an instruction that cancels the request matching a specific tag number among the requests stored in the queue. In this manner, by removing the request from the "queue" when the prediction is off, unnecessary access by the execution-pending request can be prevented, which would be desirable.

[0016]

The auxiliary storage unit of the present invention is characterized as comprising: a storage medium in which data is stored; cache memory wherein anticipated read requests are executed and data is read from the storage medium and temporarily accumulated therein; and an interface that receives anticipated read requests, including tag number information, to the cache memory from an external controller; and a controller that holds a plurality of pre-execution requests in a queuing state in which the requests are identified by the tag number, in response to the anticipated read requests received from the interface; wherein the interface received from the external controller is a cancellation signal specifying a tag number indicating the specific request whose execution is to be canceled, and wherein the controller parses the cancellation signals, and deletes the request associated with the specified, specific tag number from the queue.

[0017]

By placing a plurality of requests in a "queue", the controller can execute the requests by changing their sequence in consideration of the efficiency of the execution sequence and other factors inside the auxiliary storage unit. In addition, the controller can easily

cancel execution-pending requests, thereby minimizing performance degradation even when the prediction by the external controller is off. By characterizing the interface as being able to transmit to the external controller information on what portion of the data being executed is valid, the external controller can accurately determine the next predicted read request to be transmitted, which would be an advantage.

[0018]

The computer unit to which the present invention is applied comprises a host that executes applications programs, cache memory, and an internal controller, as well as an external storage unit that reads and writes data based on access requests from the host and a controller that controls the external storage unit; wherein the controller outputs to the external storage unit a read-ahead request on the data for which a subsequent access request is anticipated to occur based on the access request issued by the host, and outputs to the external storage unit a cancellation signal that cancels the read-ahead request.

[0019]

The read-ahead request can be characterized as being a request for the immediate execution of data read from the storage medium of the external storage unit into the cache memory; and the cancellation signal can be a signal for the cancellation of the command being executed by virtue of the request. Further, the read-ahead request is a request for a command that is managed by the internal memory under a condition in which an identification number is assigned to the command; and the cancellation signal is a signal that specifies the specific identification number to be cancelled from among the commands that are managed by the internal memory. Also, the external storage unit can be characterized as analyzing the cancellation signal output from the controller and canceling the read-ahead

request, as well as outputting to the controller information on final effective data indicating what portion of the data written to the cache memory is valid. In this manner, the controller can effectively use a part of the data for canceled requests, which would be desirable.

[0020]

The auxiliary storage unit control method related to this invention can be characterized as comprising a step of receiving commands issued by an application; a step of analyzing the flow of commands received from the application; a step of determining, based on the flow of commands that has been analyzed, whether a speculative command that indicates the data to be read ahead is needed; a step of issuing the speculative command to the auxiliary storage unit if it is determined that a speculative command is needed; a step of validating the speculative commands that have been issued; and a step of issuing a cancellation command to the auxiliary storage unit if it is determined that, as a result of the validation, superfluously issued speculative commands are pending.

[0021]

The speculative command can be characterized as including a request for the immediate execution of the reading of data from the medium of the auxiliary storage unit, and the cancellation command can be characterized as canceling the speculative command when the speculative command is being executed. In this manner, requests being executed can be canceled as appropriate, which is desirable. In addition, the speculative command can be characterized as including requests that are internally held as a queue in the auxiliary storage unit, and the cancellation command can be characterized as selecting a specific command from among the commands held in the queue by the speculative command and deleting it from the queue.

Further, the speculative command can be characterized as including requests that are held through the specification of a tag number, and the cancellation command can be characterized as specifying a tag number to be canceled. In this manner, speculative commands can be canceled by a simple instruction specifying a tag number, for the tagged-queue requests that are held as a "queue" in the auxiliary storage unit. In addition, the control method can be characterized as further comprising a step of receiving information on the final effective data from the auxiliary storage unit upon completion of cancellation processing. In this manner, for example, previously accessed data can be used effectively, thereby improving overall system performance, which is desirable.

[0022]

[Modes of Embodiments of the Invention] Embodiment Mode 1

Fig.1 is a schematic diagram of the computer unit (computer system) of Embodiment Mode 1. As indicated in the figure, the computer unit on the side of host 10 comprises a CPU 12, a first bridge circuit 13, and main memory 14. The CPU 12 is connected to the first bridge circuit 13 through an external bus 11a. Connected to the first bridge circuit 13 through an external bus 11b is the main memory 14. Also connected to the first bridge circuit 13 is an expansion bus 15, which is used to expand peripheral devices. Installed on the first bridge circuit 13 are, for example, a CPU interface, a memory controller, a clock generator, a PCI (Peripheral Component Interconnect bus) interface, and other functions, which are not shown in the figure. Applications programs are executed by the CPU 12 based on programs stored in the main memory 14. Stored in the main memory 14 are device driver programs, which provide read/write instructions to an HDD unit 22 (more on this later) independent from the applications programs.

[0023]

Connected to the expansion bus 15 is an HDC (Hard Disk Controller) card 21. Connected to the HDC card 21 are an HDD (Hard Disk Drive) unit 22 as an auxiliary storage unit (external storage unit) through bus 20a and a CD-ROM unit 23 through bus 20b. The HDC card 21 functions as an external control unit for the HDD unit 22 and the CD-ROM unit 23. In particular, it is used to expand control functions on the HDD unit 22 and for accommodating other add-on HDD units. Provided in the HDD unit 22 is disk cache 54, which is configured to cache anticipated data in advance so as to minimize time-consuming access to the medium. In addition, in this Embodiment Mode, the bus 20a is comprised of an ATA (AT Attachment) bus, and the bus 20b is comprised, for example, of an ATA/ATAPI (ATA Packet Interface) bus.

[0024]

Fig. 2 is a schematic diagram of the HDD unit 22. As indicated in the figure, for the drive mechanism, the HDD unit 22 is comprised of a magnetic disk 41, which, acting as a storage medium, stores data; and a spindle motor 42, which spins and drives the magnetic disk 41. The magnetic head 43 reads and writes data from and to the magnetic disk 41. The head arm 44, equipped with a magnetic head 43 at the tip, travels above the recording surface of the magnetic disk 41. The actuator 45, while supporting the head arm 44, spins and drives the head arm 44. By means of these components, the magnetic head 43 is configured in such a way that it can move over the magnetic disk 41 in a roughly radial direction so that it can access any position on the recording surface of the magnetic disk 41.

[0025]

The drive mechanism comprised of magnetic disk 41, spindle motor 42, magnetic head 43, and actuator 45 is controlled by the control circuit 50. The control circuit

50 is comprised of an HDC (Hard Disk Controller) 51, control memory 53, disk cache 54, and a host I/F 55. These components are interconnected through a bus 56.

[0026]

The HDC 51, acting as the internal controller for the HDD unit 22, provides overall control on the HDD unit 22, governed by the control program and control data stored in the control memory 53. The HDC 51 performs servo control and computational processing for error control during data read/write operations. By these operations, the HDC drives the spindle motor 42 and the actuator 45, and executes read/write operations by using the recording and playback heads in the magnetic head 43. In addition, the HDC 51 holds in the disk cache 54 a part of the data to be recorded on the magnetic disk 41 as a recording medium; it also performs controls for the read-ahead of a part of the data stored on the magnetic disk 41 and for the holding thereof in the disk cache 54.

[0027]

Stored in the control memory 53 are a control program that is executed by the HDC 51 and control data that is used by the control program. The disk cache 54 has the functions as cache memory of temporarily storing the write data to be recorded on the magnetic disk 41 and temporarily storing the read data that is read from the magnetic disk 41. The disk cache 54 is comprised, for example, of DRAM, which has a storage capacity on the order of several MB to tens of MB. The host I/F 55 is an interface circuit that sends and receives data and commands to and from the HDC card 21.

[0028]

From the HDC card 21, which is an external controller, the host I/F 55 receives, as information on the data to be read-ahead, several types of requests that are anticipated to be read in the future. These requests are broadly

divided into non-queue requests, which are executed immediately on the spot, and tagged queue requests, which are tag-assigned, and not immediately executed requests. Upon receipt of a non-queue request, the HDC 51 immediately begins a reading operation on the magnetic disk 41 and stores the data that has been read to the disk cache 54. On the other hand, when receiving a tagged queue request, the HDC 51 temporarily holds a plurality of requests in the form of a "queue". The sequence in which the requests are held in the "queue" is determined so that they can be executed internally with minimum latency. The reading of the plurality of retained requests is executed with adequate care for the efficiency of execution sequence and through a change in sequence. In addition, the host I/F 55 receives cancellation requests from the HDC card 21. When a command being executed is canceled, valid data is created by the HDC 51, and the effective data is issued to the HDC card 21 through the host I/F 55.

[0029]

Fig. 3 is a schematic diagram of the HDC card 21. The HDC card 21 is comprised of an HDC 31, control memory 33, I/O ports 35, 36, and a host I/F 37. As such, the HDC card functions as an external controller for the HDD unit 22. The HDC 31, the control memory 33, the disk cache 34, and the host I/F 37 are interconnected through the bus 38c. The I/O port 35 is connected to the HDC 31 through the bus 38a, and the I/O port 36 is connected to the HDC 31 through the bus 38b. The HDC 31 controls the entire HDC card 21 based on the control programs and control data that are stored in the control memory 33. The HDC 31 also performs controls for the retention of a part of the data to be stored in the disk cache 34.

[0030]

Stored in the control memory 33 are the control programs that are executed by the HDC 31 and the control data that is used by the control programs. The disk cache

34 is provided with the functions as cache memory that temporarily stores the write data that is to be stored in the HDD unit 22 and temporarily stores the data read from the HDD unit 22. The disk cache 34 is the higher-level cache memory for the disk cache 54 that is provided in the HDD unit 22 shown in Fig. 2; the cache memory having a memory capacity on the order of several MB to tens of MB, comprised of DRAM, for example. Connected to the I/O port 35 is the HDD unit 22 as an auxiliary storage unit through the bus 20a. Connected to the I/O port 36 is the CD-ROM unit 23 through the bus 20b. The host I/F 37 connected to the expansion bus 15 is an interface circuit that sends and receives data to and from the host side to which it is connected.

[0031]

The HDC card 21 sends and receives settings information to and from the HDD unit 22 when the computer unit (computer system) is started, and performs initialization on the HDC card 21 based upon the settings information that is exchanged. In the present Embodiment Mode, the prediction of requested data from the host, conventionally performed by the HDD unit 22 on a stand-alone basis, is executed by the HDC card 21. With this configuration, the accuracy of prediction is improved while at the same time detailed predictions through the detailed parsing of patterns of requests is permitted. Consequently, execution requests issued by applications running on the host side 10 are not grouped or re-sorted. This invention is thus configured so that the native form, that is, the "genuine access requests", are received from the host I/F 37. Also, this invention is configured in such a way that speculative requests based upon a true "prediction" are issued from the I/O port 35 to the HDD unit 22, and cancellation signals (more on this later) are issued.

[0032]

In the following, we represent the functions of the HDC 31 in terms of blocks. As shown in Fig. 3, the HDC 31 is comprised of an access request tracing unit 61, a speculative request determination unit 62, a speculative request validation unit 63, and a cancellation request determination unit 64. The access request tracing unit 61, receiving commands (access requests) issued by an application, stores the requests, then it parses the tracks and flow of the commands to trace them. The speculative request determination unit 62, parses the requests traced by the access request tracing unit 61 and actively determines the command for which a request is likely to be made subsequently. The speculative request validation unit 63 accumulates, parses, and validates the speculative requests (speculative commands) that have previously been issued to the HDD unit 22. The cancellation request determination unit 64 is configured in such a way that, upon receiving the results of validation by the speculative request validation unit 63, it determines the speculative request that has already been issued and that is to be canceled. Upon receipt of this determination, a cancellation command is issued.

[0033]

A description follows of the flow of issuance of speculative commands and cancellation commands in this Embodiment Mode with references to Figs. 1 to 4. Fig. 4 illustrates exchanges performed by three components, the applications program, the HDC card 21, and the HDD unit 22, in terms of the action of the HDC card 21. First, the HDC 31 in the HDC card 21 waits for the issuance of a command by the applications program on the host 10 side (Step 101). Upon detecting the issuance of a command from the application (Step 102), the HDC card parses the flow of commands from the application that has occurred up to that point (Step 103). In other words, the access request tracing unit 61 of the HDC 31 performs predictions by continuously monitoring multiple true access request

commands that have been received. If no commands are issued by an application in Step 102, the HDC card waits for issuance of a command by the application in Step 101. After Step 103, the speculative request determination unit 62 determines whether a speculative command is needed (Step 104). Among the techniques for making such a prediction is one in which several access request patterns are stored and a determination is made as to whether a traced true access request matches any of the stored patterns. In the present Embodiment Mode, it suffices to use prediction techniques employed in the prior art. For example, predictions using a data read algorithm based on read-ahead called "Look Ahead" can also be used. When the issuance of a speculative command based on such a prediction is required, the HDC 31 issues a speculative command to the HDD unit 22 through the I/O port 35 and the bus 20a (Step 105). If it is determined in Step 104 that the issuance of a speculative command is not required, control directly shifts to Step 106 (more on this later).

[0034]

The HDC 31, after issuing a speculative command to the HDD unit 22, uses the speculative request validation unit 63 to validate previously issued speculative commands (Step 106). The cancellation request determination unit 64, by this validation, determines whether superfluously issued speculative commands exist (Step 107). If no superfluous speculative commands exist, the step returns to the waiting for issuance of a command from the application, which is Step 101. If a superfluous, issued speculative command exists, a cancellation command is issued to the HDD unit 22 through the I/O port 35 and the bus 20a (Step 108). After that, control returns to the waiting for issuance of a command by the application, which is Step 101, and the invention is configured to repeat the flow of issuance of speculative commands and cancellation commands, as described in the foregoing.

[0035]

Here, non-queue requests and tagged queue requests are issued as speculative commands, as described above. Speculative commands that are non-queue requests are subject to the execution of immediate read at the HDD unit 22. Consequently, a cancellation command must be issued under the condition in which the bus 20a is being used. The present Embodiment Mode is configured so that, in response to these cancellation requests that are being executed, cancellation is applied to the HDD unit 22 using a function similar to the so-called software reset. On the other hand, if a request being executed is canceled, it would be desirable, from the standpoint of the effective use of a part of the data associated with the canceled request, to be able to determine what portion of the data being executed is valid data. In consideration of this point, the present Embodiment Mode is configured so that the LBA (Logical Block Address) of the last valid data is returned to an empty register and to the HDC card 21 so that the HDC 31 can recognize what portion of the data is valid data.

[0036]

After a tagged queue request is issued, the request that is actually executed by the HDD unit 22 is always in a state in which it is recognized by the HDC 31. The speculative request validation unit 63 validates speculative commands in cognizance of these executed requests. When canceling an execution-pending tagged queue request, the cancellation can be made by specifying a tag number. In this manner, a specific tagged queue request can be canceled from a plurality of requests that are in the "queue" in the HDD unit 22. Similarly, if a tagged queue request is being executed, cancellation on it can be applied by a method similar to the non-queue request cancellation method. In this case, information on what portion of the data is valid is issued from the HDD unit

22 to the HDC card 21, in a method similar to that described above.

[0037]

A detailed description now follows of commands that are used in the present Embodiment Mode, with references to specific examples. Figs. 5 (a), (b) show an example of a read command under the ATA interface standard, as used in the present Embodiment Mode. Fig. 5 (a) is a non-queue request. Fig. 5 (b) is the read DMA (Direct Memory Access) command used in tagged queue requests. The command block includes feature, sector count, sector number, cylinder low, cylinder high, device/head, and command registers, all of which are 8-bit registers. All ATA operations are performed by the writing of a command code into the command register. In the present Embodiment Mode, which employs the LBA (Logical Block Address) method, the LBA method is selected wherein the first 2 bits from the most significant position among the 4 high bits of the device/head register are set to "1" (High). To this end, 3 bytes in the sector count, cylinder low, and cylinder high registers, and the lower 4 bits of the device/head register, are used to specify an LBA.

[0038]

Figs. 6 (a), (b), (c) show registers under the ATA interface standard used in the present Embodiment Mode, as well as extended commands. The present Embodiment Mode is configured so that previously issued speculative commands can be canceled using these registers and commands. Here, Fig. 6 (a) is an 8-bit device control register wherein writing SRST=1 to "b2" in the lower 3 bits causes all devices to be reset. The present Embodiment Mode is configured so that only a command being executed is canceled by using prescribed empty bits. In addition, Fig. 6 (b) is the NOP command that has been modified for the present Embodiment Mode. Whereas the conventional NOP command can cancel all queued commands, it cannot cancel a

specific command. In view of this fact, the present Embodiment Mode is configured so that bits 7 to 3, which are the high 5 bits of the sector count register, are used to specify only the tag number to be canceled from the queue to enable the HDD unit 22 to cancel only superfluous access requests. Fig 6 (c) shows the content of the features of the NOP command shown in Fig. 6 (b). Code "00h" specifies the action to cancel all commands. Code "02h" indicates the action to cancel a specified tag. The other codes specify no action.

[0039]

A description follows of an example of access using the present Embodiment Mode with references to Figs. 7 to 9. Fig. 7 is a schematic diagram of an access example. Fig. 8 is an example of a command in the non-queue request case in Fig. 7. Fig. 9 is an example of a command in the tagged queue request case in Fig. 7.

[0040]

First, we explain an access example using Fig. 7. In Fig. 7 , first, application ■ requests the reading of [LBA: blocks 0 to 10]. In the next action, application ■ requests [LBA: blocks 100 to 10]. And in the next action, application ■ requests [LBA: blocks 10 to 10], as illustrated in the figure. The access request tracing unit 61 of the HDC 31 shown in Fig. 3 traces these ■■■ requests. Based on these requests, the speculative request determination unit 62 of the HDC 31 predicts the next arrival of an [LBA: from 110] request. The figure shows a condition where, based on this prediction the ■ speculative command [LBA: 50 blocks from 110] has been determined. In addition, the figure shows the occurrence of an ■ [LBA: 10 blocks from 20] read request from application contrary to the prediction.

[0041]

We now explain an example of a non-queue command issued by the HDC card 21 to the HDD unit 22 in the access example of Fig. 7, with reference to Fig. 8. Fig. 8 (a) is the ATA command (Fig. 5 (a)) issued to the HDD unit 22 when an ■ [LBA: 10 blocks from 0] read request is made by application of Fig. 7. All bits in the LBA register are "0", and the sector count register indicates a block count of "10". The command register, which is set to "C8h", indicates "Retry enabled". In Fig. 8 (b), there is a request ■, i.e., an [LBA: 10 blocks from 100] request from an application, resulting in the issuance of an LBA "100" and a sector count "10", to the HDD unit 22. In Fig. 8 (c), there is a request ■, i.e., an "LBA: 10 blocks from 10" request from an application, resulting in the issuance of an LBA "10" and a sector count "10" to the HDD unit 22.

[0042]

Fig. 8 (d) illustrates the case where, based on requests ■■■ from the application, an [LBA: from 110] request is predicted, and an [LBA: 50 blocks from 110] request is issued to the HDD unit 22 as a speculative command ■. An LBA "110" and a sector count "50" are issued to the HDD unit 22. As may be clear from these command examples, the HDD unit 22 does not discriminate whether a given command is a command based on a true access request or a speculative command. In Fig. 8 (e) shows the situation where an [LBA: 10 blocks from 20] request is made from the application, contrary to the prediction, and accordingly, a command to cancel the request ■ is issued to the HDD unit 22. In this case, *bit 3*, which is the 4th low bit, for example, in the device control register shown in Fig. 6 (a) is set to 1. Thus, the present Embodiment Mode uses empty bits in the device control register and cancels commands that are being executed in an image where, as processing, software reset is issued during the execution of the command. In this manner, even when the bus 20a shown in Fig. 3 is being used, the HDD unit 22 can be enabled to cancel commands that are being executed.

[0043]

When a command being executed is canceled, the present Embodiment Mode is configured so that the HDD unit 22 informs the HDC card 21 what portion of the data that was executed is valid. Specifically, it suffices to use an interface similar to the read command used in the non-queue request shown in Fig. 5 (a), for example, to insert the LBA for the last valid data into the register, and to provide notification to the HDC card 21. This configuration enables the HDC card 21 to effectively use a part of the data associated with the canceled request, thereby improving overall performance of the system. It should be noted that any interface can be chosen, and valid data can be reported on by employing a method that assigns the LBA for the last valid data to any empty register. Subsequently, as illustrated in Fig. 8 (f), the above request ■, [LBA: 10 blocks from 20], from the application is issued to the HDD unit 22.

[0044]

Next, we explain an example of a tagged queue request command that is issued from the HDC card 21 to the HDD unit 22, in the access example shown in Fig. 7, with reference to Fig. 9. Fig. 9 (a) is the ATA command (Fig. 5 (b)) that is issued to the HDD unit 22 when an [LBA: 10 blocks from 0] request is made from application ■ of Fig. 7. The feature register indicates a block count "10" as a sector count. In addition, a tag number is represented using the high 5 bits (bits 3 to 7) of the sector count register, and tag number 1 is indicated by the code "08h" in which Bit 3 is set to 1. The bits in the LBA register are all "0". The command register indicates value "C7h", which indicates that the command is a Read DMA Queue command. In Fig. 9 (b), there has been a request ■, i.e., an [LBA: 10 blocks from 100] request from the application; consequently, the code "10h", representing a feature "10" and a sector count indicating a tag number having the

value "1" in bit 4, indicates a tag number 2, and an LBA "100" is issued to the HDD unit 22. In Fig. 9 (c), there has been a ■ request, i.e., an [LBA: 10 blocks from 10] request from the application, and a tag number 10, and LBA "10" indicated by a feature "10" and a sector count "18h" are issued to the HDD unit 22.

[0045]

Fig. 9 (d) shows the case where, based on a ■■■ request from the application, the arrival of an [LBA: from 110] request is predicted, and an [LBA: 50 blocks from 110] request as a speculative command ■ is issued to the HDD unit 22. A tag number 4, and an LBA "110", based on a feature "50" and a sector count "20h", are issued. Upon receipt of this command, the HDD unit 22 does not discriminate it in terms of a genuine request from an application or a speculative command issued by the HDC card 21, and holds it as a similar request in the "queue".

[0046]

In Fig. 9 (e), there has been an [LBA: 10 blocks from 20] request from the application, contrary to the prediction; consequently, a command to cancel the ■ request is issued to the HDD unit 22. In this case, instructions are issued using the extended NOP command that was shown in Fig. 6 (b). Specifically, the feature register is set to "02h", as shown in Fig. 6 (c), and it is indicated as a special item that the command is a request to be canceled through the selection of a queue. In addition, to specify the queue to be canceled, the sector count register is used to indicate a tag number 4, which is the code "20h". Also transmitted to the command register is the code "00h", indicating that the command is a NOP command. Subsequently, as illustrated in Fig. 9 (f), the above request ■ from the application, [LBA: 10 blocks from 20], with a tag number 5 and with a sector count "28h", is issued to the HDD unit 22.

[0047]

It should be noted that, as a method for the cancellation of queues that are subject to a tagged queue request, the method can be configured in such a way that it cancels the command being executed if the command is being executed. In this case, in an approach similar to the non-queue request cancellation explained in Fig. 8 (e), the command being executed can be canceled in an image where software resetting is issued. On the other hand, if a command being executed is canceled, the LBA for the last valid data can be reported to the HDC card 21, so that the execution data up to the cancellation point can be used effectively, which would be an advantage. In such a case, a reporting method similar to the method described above may be employed.

[0048]

Thus, the present Embodiment Mode permits a part of the processing, hitherto performed by the HDD unit 22, to be handled by the HDC card 21, so that the HDC card 21, based upon genuine requests from applications, can develop sophisticated, detailed predictions. In particular, when the accuracy of prediction must be improved through the addition of other HDD units, such prediction and command issuance functions can be provided in the HDC card 21, which is comprised of expansion boards, to permit sophisticated controls with minimum changes during a functional expansion.

[0049] Embodiment Mode 2

Whereas Mode 1 of Embodiment Mode 2 controlled the issuance of commands to the HDD unit 22 through the use of the HDC card 21, represented by an expansion board, for example, Embodiment Mode 2 provides the functionality of the HDC card 21 on the host side, such that the command issuance/control functionality with respect to the HDD unit 22 is provided in the main memory, which is installed

on the motherboard, for example. Functions similar to Embodiment Mode 1 are assigned similar codes, and a detailed description thereof is omitted herein.

[0050]

Fig. 10 is a schematic diagram of the computer unit (computer system) used in Embodiment Mode 2. As indicated in the figure, in this computer unit a second bridge circuit 71 is connected to the expansion bus 15 on the host side. Peripheral circuits, not shown in the figure, can be connected to the second bridge circuit 71 through a bus 75. Connected to the second bridge circuit 71 is an HDD unit 22 through a bus 4a. Also, a CD-ROM unit 23 is connected through a bus 4b. The bus 75 may be comprised of an ISA (Industry Standard Architecture) bus, for example. In the present Embodiment Mode, the bus 4a is comprised of an ATA bus, and the bus 4b is comprised of an ATA/ATAPI bus, for example. The second bridge circuit 71 is provided to permit the connection of peripheral devices to an expansion bus subject to different standards than the expansion bus 15. The second bridge circuit 71 has functions such as a PCI bus interface, an ISA bus interface, a system I/O controller, and a DMA controller.

[0051]

The main memory 80 sends and received data to and from the HDD unit 22 according to instructions given by the CPU 79. Stored in the main memory 80 is a device driver 81, which is a program. The device driver 81 provides read/write directives to the HDD unit 22, independent from the applications program. As one of its functions, the device driver 81 includes an access request tracing unit 82 that parses and traces access requests that are directed to the HDD unit 22. Also provided in the device driver is a speculative request determination unit 83 that actively determines the command for which a request may be made in the future, by parsing the requests traced by the speculative request determination unit 82. Also provided

in the device driver is a speculative request validation unit 84 that accumulates and parses speculative requests after a speculative request (speculative command) is issued to the HDD unit 22 based on the determination made by the speculative request determination unit 83. Also provided is a cancellation request determination unit 85 that determines the speculative request to be canceled among the previously issued requests, based on the validation performed by the speculative request validation unit 84.

[0052]

The device driver 81 having these functions issues speculative commands and cancellation commands to the HDD unit 22, similar to the HDC 31 of the HDC card 21 described in Embodiment Mode 1. Specifically, these commands issued by the device driver 81 are issued to the HDD unit 22 through the first bridge circuit 13, the second bridge circuit 71, and the bus 74a. The HDD unit 22 parses these commands, similar to Embodiment Mode 1. When a tagged queue request is received, the HDD unit internally retains the access request in the form of a queue", and cancels the queue identified by the tag in accordance with the cancellation request. The HDD unit is also configured such that, when a cancellation command is issued with respect to a command being executed, it immediately cancels the execution, and returns an LBA to the device driver 81, indicating what portion of the data being executed is valid data. Thus, Embodiment Mode 2 permits the host side to execute controls through the use of the HDD unit 22, without requiring the use of expansion boards, such as the HDC card 21. In this manner, if the system can be designed by taking disk unit (HDD unit 22, etc.) expansion potential into consideration, the cost can be minimized compared to the situation where separate expansion boards must be installed.

[0053]

As described in the foregoing, Embodiment Modes 1 and 2 permit the parsing of access requests through the parsing of genuine requests from applications, and in this manner, "true prediction" with improved prediction accuracy can be performed. Further, because predictions are made in the form of delegating a part of the processing hitherto performed by the HDD unit 22, detailed, large-scale predictions can be performed without imposing a large overhead on the internal controller of the HDD unit 22, thereby permitting the effective utilization of the CPU power and substantial enhancement in prediction adaptability. Further, the Embodiment Modes permit the issuance of commands to the HDD unit 22 in terms of non-queue and tagged queue requests, and can provide a mechanism for canceling these access requests.

[0054]
[Effects of the Invention]

As described above, the present invention can improve the accuracy of prediction through the performance of detailed analysis and prediction by an external controller on genuine requests from applications, and through the issuance of read-ahead requests to auxiliary storage units, such as HDDs. In addition, the provision of a mechanism whereby in-execution or execution-pending requests can be canceled from the external controller minimizes performance degradation in the event of a failed prediction, and thus improves overall system performance.

4. BRIEF DESCRIPTION OF DRAWINGS

【Fig. 1】 Schematic diagram of the computer unit (computer system) in Embodiment Mode 1.
【Fig. 2】 Schematic diagram of HDD unit 22.
【Fig. 3】 Schematic diagram of HDC card 21.
【Fig. 4】 Flowchart depicting the flow of speculative

command and cancellation command issuance in the Embodiment Mode.

【Fig. 5】 (a), (b): Example of the read command under the ATA interface standard used in the Embodiment Mode.

【Fig. 6】 (a), (b), (c): Example of registers and extended-standard commands under the ATA interface standard used in the Embodiment Mode.

【Fig. 7】 Schematic diagram of an example of access.

【Fig. 8】 Description of an example command in the case of non-queue requests.

【Fig. 9】 Description of an example command in the case of tagged queue requests.

【Fig. 10】 Schematic diagram of the computer unit (computer system) in Embodiment Mode 2.

[Numerics in Figures]

10 … host side
11a, 11b … external bus
12 … CPU
13 … first bridge circuit
14 … main memory
15 … expansion bus
20a, 20b … bus
21 … HDC card
22 … HDD unit
31 … HDC
33 … control memory
34 … disk cache
35, 36 … I/O port
37 … host I/F
38a, 38b, 38c … bus
50 … control circuit
53 … control memory
54 … disk cache
55 … host I/F
61 … access request tracing unit
62 … speculative request determination unit

63 … speculative request validation unit
64 … cancellation request determination unit
70 … host side
71 … second bridge circuit
74a, 74b … bus
79 … CPU
80 … main memory
81 … device driver
82 … access request tracing unit
83 … speculative request determination unit
84 … speculative request validation unit
85 … cancellation request determination unit

(Continued from the front page)

Applicant: 599152728
Promise Technology Inc
1460 Koll Circle, San Jose 95112
United States

Inventors: Atsushi Kanamaru
1 Kirihara-cho, Fujisawa-shi, Kanagawa
IBM Japan, Fujisawa office

Hideo Asano
1 Kirihara-cho, Fujisawa-shi, Kanagawa
IBM Japan, Fujisawa office

Akira Kihashi
1 Kirihara-cho, Fujisawa-shi, Kanagawa
IBM Japan, Fujisawa office

Koichi Kushida
1 Kirihara-cho, Fujisawa-shi, Kanagawa
IBM Japan, Fujisawa office

Takahiro Saito
1 Kirihara-cho, Fujisawa-shi, Kanagawa
IBM Japan, Fujisawa office

Chichen UU
1150 Lammy Place, Los Artos, California
94024, United States

Kelvin Kao
4088 Ruppell Ave, San Jose, California
95135, United States

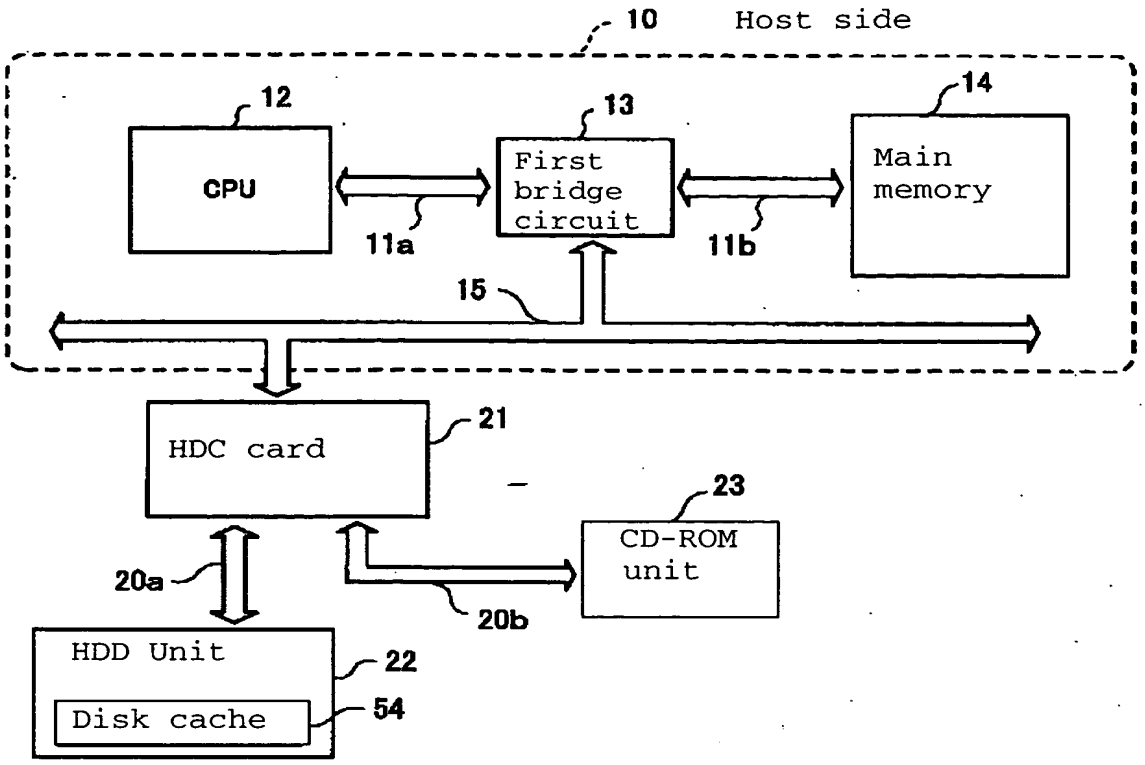

10
Host side
12
CPU
13
First bridge circuit
14
Main memory
11a
11b
15
HDC card
21
23
CD-ROM unit
20a
20b
HDD Unit
22
Disk cache
54

Fig. 1

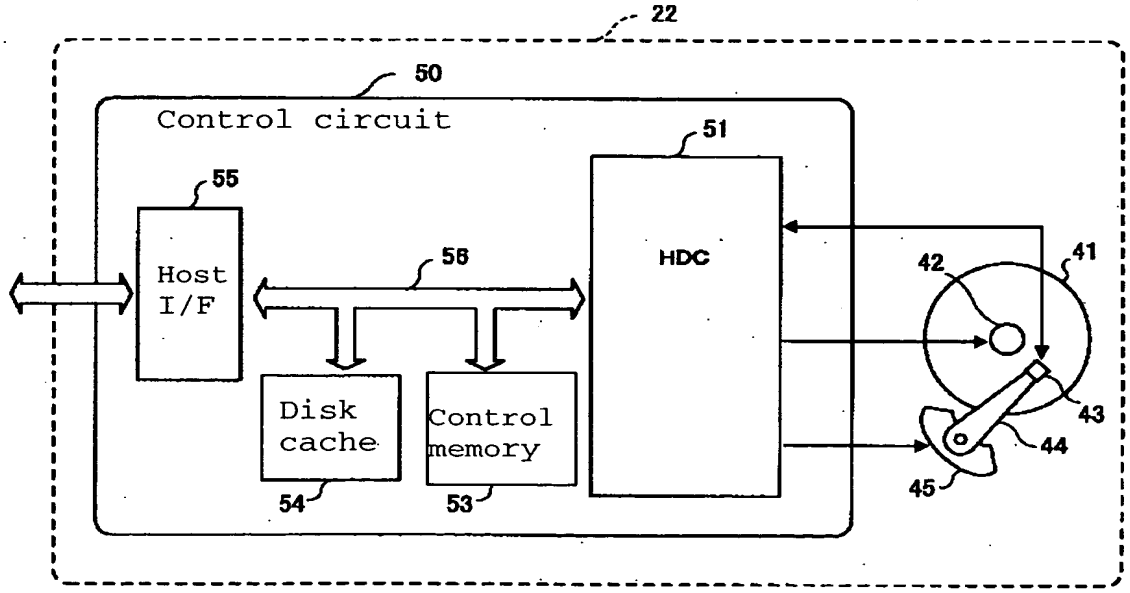

22
50
Control circuit
51
55
Host I/F
56
HDC
42
41
Disk cache
Control memory
43
44
45
54
53

Fig. 2

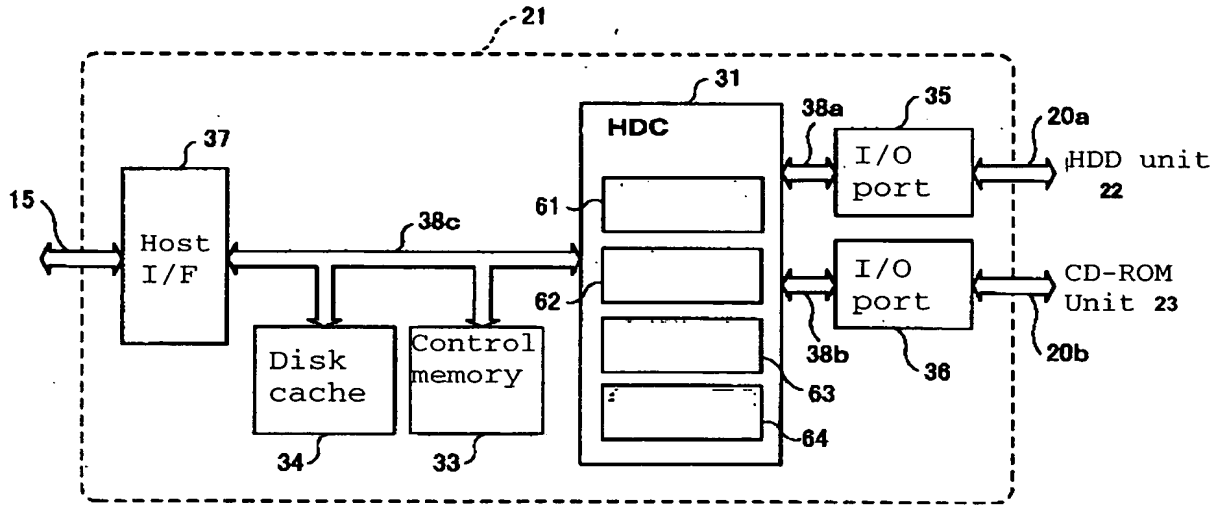


Fig. 3

61 … Access request tracing unit
62 … Speculative request determination unit
63 … Speculative request validation unit
64 … Cancellation request determination unit

The flow of issuance of speculative commands and cancellation commands

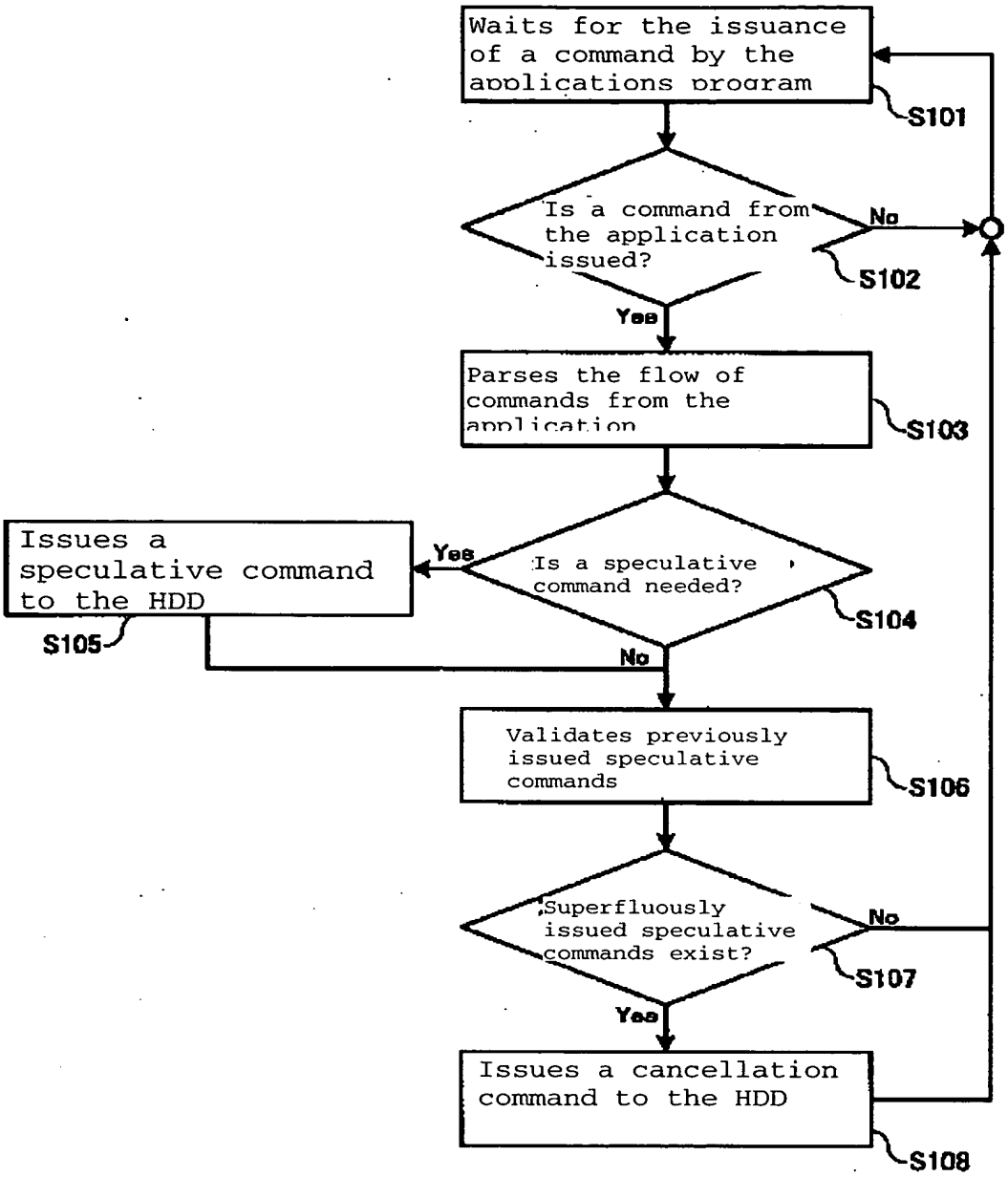


Fig. 4

Read Command Used for a Non-Queue Request

Register	7	6	5	4	3	2	1	0
Features	na							
Sector Count	Sector count							
Sector Number	Sector number or LBA							
Cylinder Low	Cylinder low or LBA							
Cylinder High	Cylinder high or LBA							
Device/Head	obs	LBA	obs	DEV	Head number or LBA			
Command	C8h or C9h							

(a)

-Read Command Used for a Tagged Queue Request

Register	7	6	5	4	3	2	1	0
Features	Sector count							
Sector Count	Tag					na	na	na
Sector Number	Sector number or LBA							
Cylinder Low	Cylinder low or LBA							
Cylinder High	Cylinder high or LBA							
Device/Head	obs	LBA	obs	DEV	Head number or LBA			
Command	C7h							

(b)

Fig. 5

Device control register

b7	b6	b5	b4	b3	b2	b1	b0
Reserved	Reserved	Reserved	Reserved	Reserved	SRST	nIEN	0

(a)

NOP command (00h)

Register	7	6	5	4	3	2	1	0
Features	Subcommand code							
Sector Count	Tag					na	na	na
Sector Number	Na							
Cylinder Low	Na							
Cylinder High	Na							
Device/Head	obs	na	obs	DEV	na	na	na	na
Command	00h							

(b)

Content of the Features Resister

Code	Description	Action
00h	NOP	Cancel all queued commands
01h	NOP Auto POLL	No action
02h	NOP Selection POLL	Cancel a specified tag
03h-FFh	Reserved	No action

(c)

Fig. 6

Schematic diagram of accesses

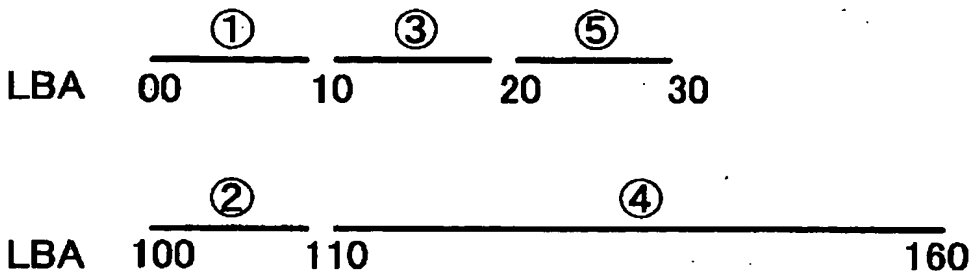


Fig. 7

Example commands for the case of non-queue requests

Register	Contents
LBA	0
Sector Count	10
Command	C8h

(a)

Register	Contents
LBA	100
Sector Count	10
Command	C8h

(b)

Register	Contents
LBA	10
Sector Count	10
Command	C8h

(c)

Register	Contents
LBA	110
Sector Count	50
Command	C8h

(d)

Register	Contents
Device Control	bit3 : 1

(e)

Register	Contents
LBA	20
Sector Count	10
Command	C8h

(f)

Fig. 8

Example command for the case of tagged queue requests

Register	Contents
Features	10
Sector Count	08h(Tag:1)
LBA	0
Command	C7h

(a)

Register	Contents
Features	10
Sector Count	10h(Tag:2)
LBA	100
Command	C7h

(b)

Register	Contents
Features	10
Sector Count	18h(Tag:3)
LBA	10
Command	C7h

(c)

Register	Contents
Features	50
Sector Count	20h(Tag:4)
LBA	110
Command	C7h

(d)

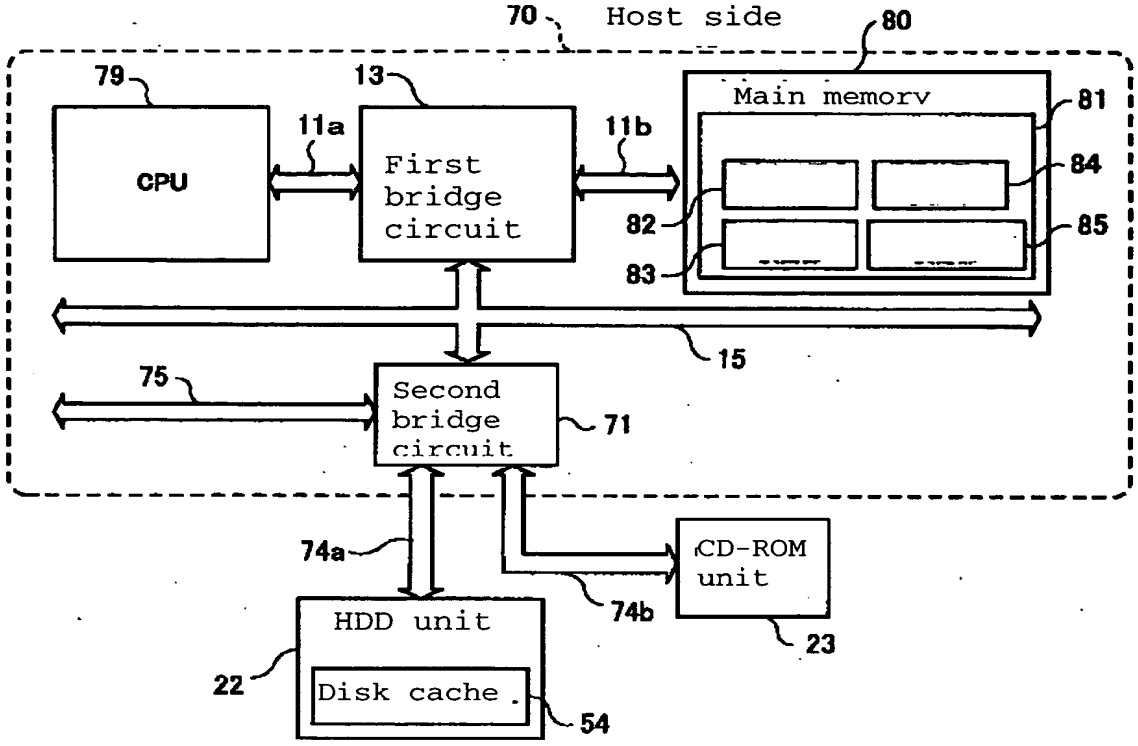
Register	Contents
Feature	02h
Sector Count	20h(Tag:4)
Command	00h

(e)

Register	Contents
Features	10
Sector Count	28h(Tag:5)
LBA	20
Command	C7h

(f)

Fig. 9



82 … Access request tracing unit
83 … Speculative request determination unit
84 … Speculative request validation unit
85 … Cancellation request determination unit

Fig. 10